DESIGN AND DEVELOPMENT OF A NEW TIME INTEGRATION FRAMEWORK, GS4-1, AND ITS APPLICATION TO SILICA PARTICLE DEPOSITION

A DISSERTATION
SUBMITTED TO THE POSTGRADUATE OFFICE
OF THE UNIVERSITY OF CANTERBURY
BY

Siti Ujila Binti Masuri

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Dr. Mathieu Sellier, Adviser

December 2012

UNIVERSITY OF CANTERBURY

This is to certify that I have examined this bound copy of a doctoral thesis by

Siti Ujila Binti Masuri

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Dr. Mathieu Sellier

Name of Supervisor

Signature of Supervisor

Date

Dedication

To my beloved hubby, lovely daughter, and loving late mother

Acknowledgments

I am heartily thankful to my supervisor, Dr. Mathieu Sellier, for all his encouragement, guidance and support throughout my PhD study at the University of Canterbury. I owe my deepest gratitude to him for he has made available his support in a number of ways. This work would not have been possible without his guidance and advice. I would also like to express my warm and sincere thanks to Prof. K. K. Tamma and Dr. Xiangmin Zhou from University of Minnesota for the opportunity to work with them and enhance my knowledge particularly in the field of time integration where my passion is.

I owe my loving thanks to my husband Mohammad Asraff Ayob and my daughter Aliya Altafunnisa who have lost a lot due to my research commitment. Without their encouragement and understanding it would have been impossible for me to finish this work.

I would also like to thank Dr. Mark Jermy, Alelign Gessese, Pavlo Kokhanenko, and Luke Sinclair from University of Canterbury, as well as Masao Shimada, Vincent Wheeler, and Andrew Hoitink from University of Minnesota for their collaboration, advice, and friendship along the way. During my research abroad in Christchurch, New Zealand, I have also met many colleagues from Malaysia for whom I have great regard, and I wish to extend my warmest thanks to Dr Azlina Yassin and Dr Azrina Md Ralib and all those who have given me their warmest friendship.

I am also very pleased to acknowledge support and funding from Mighty River Power, New Zealand, under research contract number E5653. Thanks are also due to computer grants from the Minnesota Supercomputing Institute (MSI), University of Minnesota.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

Abstract

Growing interest in the simulation of first order transient systems, typical of those encountered in transient heat conduction, flow transport, and fluid dynamics, has prompted the development of a variety of time integration methods for solving these systems numerically. The primary contribution of this thesis is the design and development of a new time integration/discretization framework, under the class of single step single solve algorithms which are the most popular, for use in such first order transient systems with computationally attractive features. These include second order accuracy, unconditional stability, zero-order overshoot, and controllable numerical dissipation with a new selective control feature which overcomes the restrictions in the existing and current state-of-the-art methods. Throughout the thesis, we demonstrate the capability and advantage of the newly developed framework, termed GS4-1, in comparison to existing methods using various types of numerical examples (both linear and nonlinear). The numerical results consistently demonstrate the roles played by the new feature in improving the numerical solutions of both the primary variable and its time derivative which is important to correctly capture the dynamics of the problems, in contrast to the existing methods without such a feature. Additionally, a breakthrough contribution presented in this thesis is the development of an isochronous integration framework (*i*Integrator), stemming from the novel relations between the newly developed GS4-1 framework and the existing GS4-2 framework (for second order dynamic systems). Such a development enables the use of the same computational framework to solve both first and second order dynamic systems without having to resort to the individual GS4-1 and GS4-2 frameworks; hence the practicality in the computational and implementation aspects. Finally, the application of the new GS4-1 framework to silica particle deposition, which is a practical problem of interest, is presented with the focus primarily on the physics of the problem. In this part of the thesis, a numerical model of the problem is presented and employed to investigate the effects of the flow and physicochemical parameters on the rate of deposition. The results of the parametric studies undertaken based on the employed numerical model enable some recommendations for the mitigation of the problem, and therefore serve as additional valuable contribution of the thesis.

Deputy Vice-Chancellor's Office
Postgraduate Office

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Co-Authorship Form

This form is to accompany the submission of any PhD thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication. A copy of this form should be included for each co-authored work that is included in the PhD thesis. Completed forms should be included at the front (after the thesis abstract) of each copy of the thesis submitted for examination and library deposit (including electronic copy).

Please indicate the chapter/section/pages of this thesis that are extracted from co-authored work and provide details of the publication or submission from the extract comes:

(1). Chapter 3 and Sections 4.1, 4.5, 5.1 and 5.2 are extracted from:
S. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. Design of order-preserving algorithms for transient first-order systems with controllable numerical dissipation. International Journal for Numerical Methods in Engineering, 88:1411-1448, 2011.

(2). Section 5.4 is extracted from:
S. Masuri, K.K. Tamma, X. Zhou, and M. Sellier. GS4-1 computational framework for heat transfer problems: Part 1 - linear cases, with illustration to thermal shock problem. Journal of Numerical Heat Transfer, 2012. to appear.

(3). Section 6.2 is extracted from:
S. Masuri, K.K. Tamma, X. Zhou, and M. Sellier. GS4-1 computational framework for heat transfer problems: Part 2 - extension to nonlinear cases, with illustrations to radiation heat transfer problem. Journal of Numerical Heat Transfer, 2012. to appear.

(4). Method described in Section 7.4 is also discussed in:
V. Wheeler, S. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. On the applicability of an isochronous integration framework for parabolic/hyperbolic heat conduction type problems. Journal of Numerical Heat Transfer, 2012. to appear.

Please detail the nature and extent (%) of contribution by the PhD candidate:
The work described in papers (1) to (3) is almost entirely Siti's (90%). Siti did the work and wrote the paper with minor input from other authors.
The contribution to paper (4) by Siti is around 30%. She contributed to the methodology.

Certification by Co-authors:
If there is more than one co-author then a single co-author vcan sign on behalf of all
The undersigned certifys that:

- The above statement correctly reflects the nature and extent of the PhD candidate's contribution to this co-authored work
- In cases where the PhD candidate was the lead author of the co-authored work he or she wrote the text

Name:	Signature:	Date:
Mathieu Sellier		29/06/2012

The following papers, based on sections of this thesis, have appeared or been accepted for publication

1. S. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. Design of order-preserving algorithms for transient first-order systems with controllable numerical dissipation. *International Journal for Numerical Methods in Engineering*, 88:1411 - 1448, 2011.

2. S. Masuri, K.K. Tamma, X. Zhou, and M. Sellier. GS4-1 computational framework for heat transfer problems: Part 1 - linear cases, with illustration to thermal shock problem. *Journal of Numerical Heat Transfer Part B - Fundamentals*, 62 (2-3):141-156, 2012.

3. S. Masuri, K.K. Tamma, X. Zhou, and M. Sellier. GS4-1 computational framework for heat transfer problems: Part 2 - extension to nonlinear cases, with illustrations to radiation heat transfer problem. *Journal of Numerical Heat Transfer Part B - Fundamentals*, 62(2-3):157-180, 2012.

4. V. Wheeler, S. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. On the applicability of an isochronous integration framework for parabolic/hyperbolic heat conduction type problems. *Journal of Numerical Heat Transfer Part A-Applications*, 62(5): 372-392, 2012.

The work described in this thesis has also been presented, based on sections of the thesis, at the following conferences

1. S.U. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. GS4-1, a new time-integrator with selective control of high frequency damping for integrating first order transient systems, *16th International Conference on Finite Elements in Flow Problems (FEF2011)*, Munich, Germany, 23-25 March 2011.

2. S. Masuri and M. Sellier. Solving time-dependent engineering problems: GS4-1, a new solver for first order system with optimal properties. *1st National Postgraduate Research Conference (1NPRC)*, Dunedin, New Zealand, 1 July 2011.

3. S.U. Masuri, V. Wheeler, M. Sellier, X. Zhou, and K.K. Tamma. A Unified Order Preserving Framework for Integrating First/Second Order Parabolic/Hyperbolic Systems. *11th US National Congress on Computational Mechanics (USNCCM11)*, Minneapolis, USA, 25-29 July 2011.

4. S. Masuri, M. Sellier, K. Brown, and M. Rock. Numerical Model of Silica Colloids Deposition. *2011 Biennial New Zealand Geochemical & Mineralogical Society (NZGeMS) and 4th Meeting of the Stable Isotope Network for New Zealand (SINNZ) Joint Conference*, Christchurch, New Zealand, 16-18 November 2011.

5. S. Masuri, K. Brown, and M. Sellier. Numerical Studies of Silica Colloids Deposition: Effects of Flow and Physicochemical Parameters on Rate of Deposition. *US-NZ Joint Geothermal Workshop 2012*, Rotoroa, New Zealand, 16-20 April 2012.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of Topic

Many problems to be solved by engineers, researchers or academicians are dynamics or transient systems. Dynamic systems mean that the solutions of the problem change with time. Among the popular examples of such systems are the so-called structural dynamic problems, which describe the response of a structure over time during and after the application of a load, a situation which frequently occurs in buildings, vehicles, aeroplanes, bridges etc. In the heat transfer field, the heat conduction problems, which describe the transfer of thermal energy between regions of an object due to differences in temperature, could be time-dependent due to an imposed change in temperature at the object boundary. This time-dependent heat conduction may also occur when a source (or sink) of heat is suddenly applied within the object and subsequently causes changes in the nearby temperatures. In fluid dynamics, which is an engineering discipline that deals with the natural science of liquids and gases (so-called fluids) in motion, most problems are described by the Navier-Stokes equation. Named after Claude-Louis Navier and George Gabriel Stokes, this system of equations, which is time-dependent in general, allows researchers in this field to handle a wide range of applications, including calculating forces on aircraft, determining the rate of flow of petroleum through pipelines, and predicting the deposition of small particles from liquid suspension onto the pipe wall.

Solving time-dependent engineering problems is not always simple. In practice or even in the academic world, engineers or researchers often have to deal with engineering problems whose equations involve multiple physics while geometries are large-scale if not complex. For such problems, it is almost always impossible to derive analytical solutions to the problems. Therefore, it has become common practice for engineers or researchers to rely on numerical methods or computational softwares to obtain such solutions. There are many commercially available codes which are currently used for solving dynamic problems. For example, simulations of structural dynamic problems can be done via ANSYS, Abaqus FEA, and the newly launched software Creo. For fluid dynamics problems, many computational fluid dynamic (CFD) softwares are available such as COMSOL, including the open sources such as OpenFOAM. Simulations of rigid multibody systems such as vehicle dynamics, analysis of wind turbines, or robot dynamics can be done via ADAMS and DADS. Researchers and engineers can also solve the problems by writing their own computational codes via programming languages such as Fortran, Matlab and C++.

Solving these problems numerically or computationally usually involves two major steps. Firstly, the whole spatial domain of the problem, as enclosed by the geometry, is broken into discrete elements. These discrete elements are interconnected such that their assembly represents the problem's actual continuous spatial domain. By doing so, one is allowed to create elemental equations that approximate the actual equation of the problem. This step, known as spatial discretization, transforms the problem's actual equation (that is complicated due to its nature of being partial differential) to a set of ordinary differential equations that is easier to solve. A number of different spatial discretization methods are available including Finite Element Method, Finite Difference Method, Finite Volume Method, and Boundary Element Method to name a few. When the problem is time-dependent, the resulting set of ordinary differential equations will be time-dependent. Some problems are dependent on time to a first order, such as parabolic heat conduction problems. Others are second order in time such as hyperbolic heat conduction problems. Once the problem's equation is written as a set of ordinary differential equations after use of any spatial discretization method, the next step in the numerical method is to solve such a set of equations using a solver or method that can give the solutions to the problem at each time

level and throughout the whole simulation period. This step is called the time discretization/integration, and the methods used for such a purpose is called the time discretization/integration methods or time discretized operators. To date, there exists in the literature a variety of different classes of time integration methods for solving time-dependent engineering problems, such as the so-called linear multi-step methods, sub-stepping methods, Runge-Kutta type methods, higher-order time accurate methods, etc.

1.2 Overview and Scope of Research

The present research is primarily focused on the issues and developments related to the time discretization part of the computational procedures involved in solving engineering dynamic systems. The spatial discretization part, on the other hand, has received almost no special attention, and well-established spatial discretization methods such as the standard Galerkin and Streamline Upwind Petrov Galerkin (SUPG) Finite Element Methods (FEM) are readily employed. Turning attention to the time discretization part, for a robust computational procedure to solve time-dependent phenomena encountered in engineering, of either first or second order systems, such as those arising in fluid dynamics, transient parabolic and hyperbolic heat conduction, structural dynamics, wave propagation, etc., time integration methods play an important role to properly integrate the consequent time continuous ordinary differential equations obtained after the spatial discretization. It has long been recongnized that it is extremely crucial for the equations to be optimally integrated by the time integration methods such that the resulting numerical dissipation, dispersion, and algorithm overshoot are only minimal. Additionally, it is important that shocks can be captured without too much dissipation, stiff problems can be solved accurately, convergence of nonlinear iteration can be achieved easily, and more importantly that the completion of the analysis can be accomplished.

Among the many classification of time integration methods, such as the linear multi-step methods, sub-stepping methods, Runge-Kutta type methods, higher-order time accurate methods, etc., probably the most popular and widely used in practical applications and commercial software are those pertaining to the class of single step single solve due to its practical use and simple computational implementation.

In this class, the system of equations is solved only once at each time level using solutions of only one previous time level (i.e., single step single solve). These characteristics of time integration methods make it probably the simplest of its kind, which subsequently require the least computational effort. Due to such advantage and convenience of this class of time integrators, we have been focusing much of our previous effort on its design and development. In this research, we again focus our developments and discussions on the time integration methods pertaining to the class of single step single solve due to the reasons cited above.

The development of a new time integration framework under the particular class of single step single solve algorithms is the primary focus of the present research. Additionally this study, which was co-funded by Mighty River Power, one of the major companies in New Zealand that operate geothermal power stations, was to be one of several studies in a series of research projects with an overarching goal of gaining better understanding of the silica particle/colloidal deposition problem, a problem that is costly yet common in the geothermal power plants. In this regard, the objective in the present study, apart from the one mentioned above, is also to develop a computational code to model the particle deposition problem based on the existing/well-founded numerical model of such a problem. Ultimately, it is hoped that the employed model can be used to gain more insights into the problem. In particular, the research sought to understand the effects of the various flow and physicochemical parameters on the rate of deposition, so that more effective measure can be developed from such a study. This particle deposition problem is dynamic in nature and is governed by partial differential equation that is first order in time. Therefore, solving such a problem requires the time integration technique that is appropriate for such a system. For this reason, it is important that the development of the new time integration framework be tailored to suit such a purpose. Therefore, of particular interest in this research is the development of a time integration framework encompassing the following scope:

1. Comprise algorithms pertaining to the class of single step single solve only (due to the advantage and convenience of such a class of algorithms as mentioned above), and

2. Taylor the framework for applications pertaining to first order (in time) systems governed by Partial/Ordinary Differential Equations (PDE/ODE) such as the heat transfer and flow transport problems (note that the Differential-Algebraic Equation (DAE) is not considered in the present study since a system governed by such an equation exhibits fundamental mathematical properties that are different from those of PDE/ODE due to the presence of the algebraic constraint. Consequently, this type of equation poses additional challenges for their numerical solutions, which requires a thorough investigation and additional attention. Therefore, this type of systems will be addressed in the future research).

For a complete development of the new time integration framework, under the scope mentioned above, the research also sought a possible way for the new framework to be naturally integrated to the existing framework for second order dynamic systems. Such an existing framework (for second order dynamic systems) has gained much of our previous effort on its development and improvements. A precise understanding of the respective frameworks, and how the two are related to each other, has led to the development of an isochronous integration framework, which enables the use of the same computational framework for solving both first and second order dynamic systems without having to resort to the individual framework; hence the practicality in the computational and implementation aspects. This, therefore, completes the overall developments in time integration technique, and therefore can be viewed as another essential part of the research and a breakthrough contribution.

Since the main focus in this research is on the design and development of a new time integration framework as described above, the most significant aspect of the research is on investigating and demonstrating the capability and advantage of the new framework in comparison to the current state-of-the-art methods. Furthermore, to be able to show consistency of the arguments, it is necessary that the developed framework be tested on various numerical examples as described throughout the thesis, which are benchmark problems with available analytical solutions,[1] allowing for direct comparisons between the performance of the existing/current state-of-the-art methods and the newly developed time integration framework. As a consequence to

[1]except for the particle deposition problem, which is a more practical complicated problem in which analytical solutions are not available

this, little attention is purposely given to the physics of the benchmark problems with the intention to primarily highlight the advantages and ability of the newly developed time integration framework in comparison to the existing and current state-of-the-art methods. Having achieved this, the discussion in the thesis is then turned to and focused on the physics of the particle deposition problem as a complement.

1.3 Research Objectives and Contributions

The primary objectives of the present research are the following:

1. Analyze the key restrictions in the existing and current state-of-the-art time integration methods to solve time-dependent engineering problems pertaining to first order transient systems and under the particular class of single step single solve algorithms.

2. Develop a new framework of single step single solve time integration algorithms for use in first order transient systems with computationally attractive features which overcomes these existing restrictions, and therefore contributes towards improving the state-of-the-art of such a class of algorithms. Such a development involves the design of the new framework and its application to various types of numerical examples (both linear and nonlinear) governed by first order (in time) partial/ordinary differential equations to demonstrate the ability and advantage of the new framework in comparison to the existing methods.

3. Investigate how the new time discretization framework can be naturally integrated with the existing framework for second order dynamic systems, hence allowing for the development of an isochronous integration framework, which enables the use of the same computational framework for solving both first and second order dynamic systems without having to resort to the individual frameworks; hence the practicality in computational and implementation aspects, and therefore completes the whole developments of the time integration method.

4. Search for available numerical models of particle deposition from laminar suspension that can be used in this study for gaining more insights into such a practical problem of interest.

5. Investigate using the employed particle deposition model the effects of the flow and physicochemical parameters on the rate of deposition, so that more effective measure to mitigate the problem can be further developed from such a preliminary study.

1.4 Thesis Outline

Chapter 1, *Introduction*, introduces the research topic and highlights the scope and objectives of the research presented in this thesis.

Chapter 2, *Literature Review*, reviews the existing and current state-of-the-art methods and highlights the key restrictions in these methods.

Chapter 3, *Design and Development of the New GS4-1 Time Integration Framework*, presents the theoretical developments leading to the design of the new GS4-1 time integration framework.

Chapter 4, *The New GS4-1 Time Integration Framework and the Relevant Details* presents the newly developed GS4-1 framework and provides further relevant details for a complete description of the new framework, including the general guidelines on how to use the new framework.

Chapter 5, *Application of GS4-1 Framework to Linear First Order Transient Systems*, demonstrates how the new framework with its computationally attractive features overcomes the restrictions in the existing and current state-of-the-art methods through applications to various linear first order transient systems.

Chapter 6, *Extension of GS4-1 Framework to Nonlinear First Order Transient Systems*, describes how the new framework, originally designed for linear systems, can be properly extended for use in nonlinear applications pertaining to first order transient systems. Application to a number of nonlinear systems consistently demonstrates the ability of the new framework in comparison to the existing/current

state-of-the-art methods for these nonlinear applications as those seen in the linear dynamic situations.

Chapter 7, *The Relations of the New GS4-1 and the Existing GS4-2 Time Integration Frameworks: Development of an Isochronous Integration Framework (iIntegrator) for First/Second Order Dynamic Systems*, discusses how the new GS4-1 framework and the existing GS4-2 framework (for second order dynamic systems) are related to each other, followed by the description of the *i*Integrator. Finally, we show its applicability and practicality through applications to dynamic thermomechanical problems involving both first and second order dynamic systems.

Chapter 8, *Application of GS4-1 Framework to Silica Particle Deposition*, presents the practical problem of interest, both from physics and computational point of view. It describes the numerical model of the problem and how the GS4-1 framework can be applied to solve such a model. Having validated the model, additionally, it illustrates the effects of the various flow and physicochemical parameters on the rate of deposition. Some recommendations for the mitigation of the problem are also given based on the results of such a parametric study. Finally, some illustrative numerical results of the deposition of silica particles are presented for completeness.

Chapter 9, *Concluding Remarks and Future Directions*, highlights the fundamental contributions of the present research and suggests ideas and directions for future research in this line of approach.

Chapter 2

Literature Review

The intention of this chapter is to provide readers an overview of the existing time integration methods that fall under the class of single step single solve algorithms. Although the scope of the research, as mentioned before, is on the development of a new time integration method that can be used for applications pertaining to first order transient systems, we recall that another essential part of the research is also on the integration of this new framework to the existing framework for the second order dynamic systems, leading to the development of an isochronous integration framework that can be used to solve both first and second order systems; hence, completing the overall developments in time integration techniques. For this reason, it is important that we provide a review of the existing and current state-of-the-art methods for both first and second order systems. This review is necessary to understand the key restrictions in these existing methods, and to understand how the new developments presented in this thesis fit and overcome these restrictions and therefore contribute to the state-of-the-art in this field. First, an overview of the single step single solve time integration methods for second order systems are given, as we have been focusing much of our previous effort on developments and improvements in this area, and then followed by the review of existing methods for the first order systems. The review of literature related to the silica particle deposition, which is the practical problem of interest, is however presented in Chapter 8 for clarity purpose.

2.1 Time Integration Methods for Second Order Dynamic Systems

For the second order dynamic systems, such as those encountered in hyperbolic heat conduction, elastodynamics, and wave propagation, etc., the existing algorithms of this classification[1] include the Newmark method [10] and its variants [11–13] as well as the Midpoint rule algorithm and its variants [14, 15]. They have similar stability, accuracy, overshoot behaviour, numerical dissipation, and numerical dispersion properties. Although these numerically nondissipative algorithms possess intrinsic energy conserving property which is of particular interest in unconstrained conservative dynamic systems, they fail to filter out the spatially induced high frequency modes. On the other hand, in damped dynamic systems, the algorithmic damping of these algorithms tends to vanish for the high frequency modes. This is a critical problem especially for large scale multi-degree of freedom dynamic problems where the semi-dicretized equation of motion may become highly stiff especially when adaptive mesh is employed in region with concentrated strain and stress. These spatially induced high frequency modes, when participating in the solution, may cause ill-conditioned matrix the solution of which may not converge or may destroy the solution accuracy, often resulting in numerical oscillations. Because of such circumstances, it has long been recognized that it is extremely important for time integration methods to possess numerical damping property to enable the completion of the entire analysis and to give accurate solutions to the problem especially stiff ones. Numerical damping, also called numerical/algorithmic dissipation, represents the dissipative effect induced by the algorithm at high frequency limit (hence sometimes also referred to as the high frequency damping) [16]. It is crucial for the algorithms to possess this property due to the shortcomings of the numerically nondissipative schemes. Additionally, it is desirable that the amount of the numerical dissipation induced by the algorithms can be controlled by the analyst depending on the needs of the problem at hand. Algorithms imposing numerical damping in controllable manner are hence referred to in the literature as the *controllable numerically dissipative algorithms*.

The first controllable numerically dissipative time integration algorithm with unconditional stability for structural dynamic problems is the Wilson-θ method [17].

[1] i.e., single step single solve time integration methods

Although this method and few others that have since been developed inherently possess controllable numerical damping that is desired to suppress the numerical oscillation resulting from the presence of spatial induced high frequency modes, they appear to suffer from overshoot behaviour. For example, as pointed out by Goudreau and Taylor [18] (see also Reference [19]), the Wilson-θ method suffers from second-order displacement overshoot and first-order velocity overshoot behaviours due to initial displacement. The Hoff-Pahl-Θ_1 method [20], which is also a numerically dissipative algorithm, suffers from first-order displacement overshoot and first-order velocity overshoot behaviours due to initial displacement as pointed out in our previous exposition [16]. Other numerically dissipative algorithms with zero-order displacement overshoot behaviour exist, but suffer from first-order velocity overshoot due to initial displacement including the Hilber-Hughes-Taylor (HHT-α) method [21, 22], the Wood-Bossak-Zienkiewicz (WBZ) method [23], and the Generalized-α method[2] [26]. Besides the issues with overshoot behaviour, these algorithms, except for the Generalized-α method, do not yield minimal algorithmic dissipation and dispersion. As compared to these algorithms, the Generalized-α method can yield minimal numerical dissipation and dispersion with the same value of the principal root at the high frequency limit (ρ_∞), hence the improvement. However, as mentioned previously, this method suffers from the first-order velocity overshoot behaviour due to initial displacement [16, 27].

The overview described above indicates that all of these aforementioned time integration methods lack in terms of overshoot property. Motivated by this, we have previously explored the development of new framework that, not only overcomes the overshoot behaviour issue associated with these algorithms, but also encompasses the broad scope of Linear Multi-Step (LMS) methods. For this purpose, and looking at the big picture of the time dimension, we have first introduced in a series of papers [16, 27–30] a new concept that describes how to design time integration methods from a generalized unified mathematical framework, namely the notion of *Algorithms by Design* for structural dynamics problems. Algorithms by design is a relatively new concept for the design procedure via which one can *apriori* tailor the design of a computational algorithm according to pre-determined desirable algorithmic

[2]The Generalized-α method is identical to the three-parameters optimal schemes presented in [24, 25].

attributes ('wish list'). Such a procedure is in contrast to classical design approach where one *aposteriori* studies the algorithmic properties resulting from an 'idea' which could be a physical based interpretation or a mathematical representation of an assumed construct. Employing the Algorithms by Design procedure for linear structural dynamic problems, it was shown previously that the so-called family of Generalized Single Step Single Solve (GS4-2)[3] time integration framework could be uniquely designed with computationally attractive features which overcome the shortcomings of existing methods. The framework provides a wide variety of choices to the analyst encompassing the entire class of LMS methods [16, 27]. It recovers most existing algorithms in the literature regardless of the original approach of how the algorithms were developed. These include numerically nondissipative methods such as the Mid-Point Rule, Newmark method, and the so-called Velocity Based Scheme [14][4] as well as numerically dissipative schemes such as the HHT-α method, WBZ method, Generalized-α method, and the like. More importantly, the framework also provides new family of optimal controllable numerically dissipative algorithms, termed U0-V0$_{optimal}$, with new desirable features. These new features, which are zero-order displacement overshoot and zero-order velocity overshoot behaviours, are clearly improvement to the previous algorithms. Additionally, such optimal algorithms yield only minimal numerical dissipation and numerical dispersion for damped dynamic systems, and also have the option of energy conserving for undamped dynamic systems.

The early development of GS4-2 framework was originally focused on the applications to linear structural dynamic systems [16, 27]. In such a situation, the Midpoint Rule and Velocity Based Scheme (both can be recovered in the GS4-2 framework with particular choices of the framework parameters) are both energy, linear and angular momentum conserving. Assessing stability for linear systems is straightforward as the stability characteristics can be readily determined by modal analysis. However, such an approach is not readily applicable for nonlinear dynamic problems [31]. Moreover, it has been routinely observed over the years that unconditionally stable algorithms

[3]formerly abbreviated as GSSSS [16]; herein referred to as GS4-2, where '2' stands for second order dynamic systems

[4]with the exception of the Newmark method, the other two are energy, and linear and angular momentum conserving for linear dynamic situations

for linear dynamic problems often fail to consistently possess this numerical property when applied to nonlinear dynamic situations. Turning our discussion to nonlinear second order dynamics problem, with particular attention to the notion of achieving unconditional stability for such systems, conservation of properties such as energy, linear and angular momentum is extremely useful for the numerical schemes to possess. For these nonlinear dynamic situations, conservation or decay of the total energy has been a criterion for assessing the stability of the algorithms as found in previous works such as Belytschko and Schoeberle [32], Hughes [31, 33], and Simo and Tarnow [34, 35] among others, since the modal analysis is not readily applicable for such situations. This has increased the research interest in developing time integration schemes that are energy and momentum conserving for nonlinear second order dynamic situations. For example, Hughes et al. [36] developed the Constraint Energy Method by extending the Trapezoid rule [10] and enforcing the energy conservation by introducing Lagrange multipliers. However, this enforced conservation of energy method does not guarantee stability as described by Kuhl and Ramm [37]. For the problem investigated, the algorithm failed to converge after a certain time point, showing that imposing energy constraint alone is not sufficient for stability in nonlinear dynamic problems. A useful conserving algorithm, the Energy Momentum Method for unconstrained conservative dynamic systems, has been developed by Simo and Tarnow [34] following the original idea of LaBudde and Greenspan [38] for particle dynamics (who first introduced the concept of enforcing energy constraint). Following the framework of Hughes et al. [36], Kuhl and Ramm [37] describe the Constraint Energy Momentum Algorithm in an ad hoc manner by combining the dissipative feature in numerically dissipative algorithms with the enforcement of not only energy, but also linear and angular momentum constraints. Although the algorithm is shown to be stable, there is no proof of consistency or order of accuracy presented; thus the convergence aspects of the algorithm is ambiguous and needs to be carefully investigated. Related efforts can also be found in previous works by other researchers including Wood and Odour [39] and Crisfield and Galvanetto [40] among others.

Despite these previous efforts in developing time integration methods that are energy and momentum conserving for undamped nonlinear structural dynamic systems, the

need for such methods remains. Motivated by this, we have previously described how the GS4-2 framework, originally developed for linear structural dynamic systems, can be properly extended for nonlinear situations via a new approach termed the *normalized time weighted residual approach* [41, 42], in contrast to all previous efforts and classical approaches. In the design of computational algorithms for nonlinear dynamic situations, we proposed that the basic primitive algorithms to march the solutions in time can indeed be the same as those for the linear dynamic counterparts (i.e., the algorithms within the original GS4-2 framework), but with careful and accurate treatment of the nonlinear terms such as the internal force and consequently, the associated algorithmic strain involved in the semi-discretized equation of motion. How to naturally achieve unconditional stability and conservation of energy, linear and angular momentum in a consistent mathematical setting using the existing GS4-2 algorithms without having to resort to enforcement of constraints *aposteriori*, and how to employ the new normalized time weighted residual approach for such a purpose are described in [41, 42], with particular application to the Saint Venant-Kirchhoff material for illustration of the basic concepts. The idea behind the new normalized time weighted residual approach is the individual weighting and normalization of each term appearing in the semi-discretized equation of motion. This relatively new approach, in contrast to the classical counterpart, inherently enables the nonlinear terms in the semi-discretized equation of motion to be treated specifically. This leads to a more accurate treatment of the nonlinear terms and thus yields a computational framework for implementation that naturally achieves conservation of energy (without resorting to additionally enforcing energy constraints as in all past works to-date), linear, and angular momentum.

Additionally, this new approach naturally allows for all possible treatments of the nonlinear term to be investigated by the analysts. For the particular problem studied (i.e., nonlinear structural dynamics with particular application to the Saint Venant-Kirchhoff material model), two treatments of the nonlinear term are possible leading to two types of time integration algorithms with different conservation properties: (1) a family of energy, linear and angular momentum conserving algorithms [41], and (2) a family of symplectic, linear and angular momentum conserving algorithms [43, 44]. For both types, the basic primitive algorithms to march the solutions in

time are those for the linear dynamic situations (i.e., the algorithms within the original GS4-2 framework [16]), but with careful treatment of the nonlinear terms, i.e., the internal force and consequently the algorithmic strain. We demonstrated in [41] the success of this new approach when employed to any basic primitive algorithm, that is energy and momentum conserving for linear dynamic situations, by applying to particular model problems with truss elements characterized by the Green strain. Numerical illustrations demonstrated that by using the GS4-2 algorithms that are energy and momentum conserving (for linear dynamic situations) to solve nonlinear dynamic problems, implemented via the concepts provided by the normalized time weighted residual approach, we were indeed able to preserve the conservation properties of these physical quantities without resorting to imposing additional constraints, namely, the energy constraint which narrows the scope of the design space for designing various conserving algorithms.

Although energy and momentum conserving algorithms have a role in a class of nonlinear dynamic applications, often, a problem that may be encountered is that the spatial discretization often leads to high frequency participation of the modes in highly stiff problems. Therefore, it must be pointed out that such situations require special treatment such as elimination of unwanted high frequencies, and/or introduction of numerical dissipation, often controllable, to enable the completion of the simulation without suffering from convergence issues in the Newton iterations often faced by nondissipative energy momentum conserving schemes. However, some of the conservation properties of the underlying physical problem will be violated. A detailed work in this line of approach can also be found in [41]. The numerical results showed that in contrast to the classical time weighted residual approach, this new normalized time weighted residual approach yields less numerical oscillations in the energy and angular momentum thus confirming the effectiveness of the developed approach. We also demonstrated that among the dissipative schemes considered, the U0-V0$_{optimal}$ is the preferred choice of the dissipative scheme since it yields the least energy dissipation and is ideal for any given starting initial conditions in the sense that it possesses zero-order displacement and velocity overshoot behaviours, both for linear [16] and nonlinear [42] second order dynamic systems.

Additionally, from the standpoint of practical engineering applications, there are classes of problems where long duration simulations are important and of interest. The traditional development of time integration operators especially those derived from the Ordinary Differential Equation (ODE) such as the Euler Forward, Euler Backward, the Newmark method and the like lack the ability to handle these long term simulations. Because of this deficiency and driven by the need to be able to handle these types of applications, there has been a research interest in developing geometric or structure preserving time integration algorithms. In this regard, the focus has been on symplectic algorithms which possess the numerical properties of preserving the phase space volume and momentum while the energy is bounded. For fixed time step algorithms, it is described in the work of Ge and Marsden [45] that there is a general division of these time integration operators into those that are energy and momentum preserving and those that are symplectic and momentum preserving. The work by Simo and Tarnow [34, 35] and by Kuhl and Crisfield [46] are the examples of those belonging to the former group whereas the latter includes the paper by Feng [47] and Veselov [48]. A more recent work in this line of approach can be found in the work of Kane et al. [49, 50] and Marsden and West [51]. Particularly for nonlinear elastodynamics applications, it is reported in the work of Noels et al. [52] that the energy preserving property and the symplectic preserving property cannot both co-exist for algorithms with a constant time step and this is the trade off one has to pay. It is also worth mentioning that the symplectic integrators have the shortcoming of blowing up the energy for a large time step size, but for a small enough time step size the energy they yield is bounded [53]. In developing symplectic time integration algorithms, there are various approaches reported in the literature (see [47–52, 54–58] among others). One approach is called the Discrete Euler-Lagrangian described in [48]. Alternatively, we have previously shown [43] that a family of symplectic and momentum preserving time integration algorithms can be uniquely recovered in the GS4-2 framework when applied to nonlinear dynamic problems via the concepts provided by the new normalized time weighted residual approach. To complete the development and to be able to solve highly stiff or large scale problems, we also demonstrated in another paper [44] how to introduce dissipation to this such that when the dissipative features are turned off, the resulting algorithms recover the design of algorithms that are symplectic and momentum preserving.

At this point, it is worthy to highlight that the review of the previous works described above is related to the second order dynamic system including our recent developments for such a system. The importance of this review will become apparent in Chapter 7, where we relate and unify the developments of such time integration methods for second order dynamic systems to those for the first order transient system, consistently, to yield the isochronous integration framework [59] that can be used for both systems. For now, we turn our attention to review the existing time integration methods for first order transient systems which is the scope of this thesis.

2.2 Time Integration Methods for First Order Transient Systems

For first order time-dependent systems, such as fluid flow problems governed by the Navier-Stokes equations, the traditional practice to solve such problems involves the use of the classical Trapezoidal family of algorithms (also known as the θ-family) which is the most popular. A number of well-known time integration methods are contained within the family including the Crank-Nicolson method [60], the Euler Backward, the Galerkin method and the Euler Forward method. Despite being popular, these methods are only first order accurate (with the exception of the Crank-Nicolson method) hence the clear disadvantage in terms of computational efficiency compared to second order accurate methods. Second order accurate methods are preferable since when we reduce the time step size the temporal errors generated by the methods reduce more rapidly than those generated by the first order accurate methods. Therefore, in terms of the order of accuracy, an analyst would prefer to use the second order accurate Crank-Nicolson method over the other methods that are only first order accurate. However, from the stability point of view, the Crank-Nicolson method suffers oscillatory decay[5] for large time step sizes. The resulting response, although bounded, is oscillatory and will not represent the true response. Therefore, this method has a clear disadvantage despite the advantage of being second order accurate method. The Galerkin method also suffers from oscillatory decay range for large time step sizes. The Euler forward difference is far worse as it may yield results that grow with time and eventually becomes unbounded, in addi-

[5] In the absence of a forcing function, the response of real systems decays with increasing time [61]

tion to being only first order accurate. In terms of stability, only the Euler Backward method yields response with a smooth decay which is non-oscillatory. However, it is worthy to recall that this method is only first order accurate. Due to this restriction, many analysts regard the selection of algorithm from the classical Trapezoidal family of algorithms as the choice between the Crank-Nicolson method and the Euler Backward method. Clearly, there is a tradeoff between order of accuracy and stability.

For nonlinear problems, the importance of the second order accuracy of time integration methods has long been recognized to ensure a higher rate of convergence of the numerical solutions to the exact solutions. For this reason, research attention has been geared towards developing methods that are not only second order accurate but also possess controllable numerical dissipative feature. For first order dynamic system, such a feature is crucial to overcome the well-known shortcoming of the second order accurate Crank-Nicolson method. The numerical solutions produced by such a nondissipative scheme often switch sign on each time step due to its zero damping property (i.e., no numerical dissipation). This results in a rather unrealistic solution behaviour which can cause non-physical instabilities and in some cases can lead to non-convergence of the nonlinear iteration. Controllable numerically dissipative algorithms with second order accuracy for integrating first order transient systems, under the class of single step single solve methods, exist [62] and constitute the current state-of-the-art method. In this method, the analyst has a precise control of the high frequency damping through a parameter ρ_∞ (hence often referred to in this thesis as the *one-parameter method*), resulting in a whole family of time integration algorithms that can be tested for the particular problem at hand. The user control parameter, ρ_∞, may be increased or decreased, i.e., ρ_∞ value may be chosen from the range of $0 \leq \rho_\infty \leq 1$ according to the needs of the problem of interest. A ρ_∞ value of one yields the nondissipative (zero-damping) Crank-Nicolson method. On the other hand, if ρ_∞ value is chosen to be zero, the method is said to annihilate the highest frequency in one step (only for linear problems) [62] resulting in an algorithm that has the same spectral stability as the Gear's method [63]. In between, the method yields algorithms with controllable numerical dissipation feature for first order transient system.

Although the one-parameter method [62] seems to overcome the well-known shortcoming of the second order accurate nondissipative Crank-Nicolson method via its user control parameter ρ_∞, the method's control of the high frequency damping via this parameter lacks flexibility. From our investigations based on the numerical results of various types of linear and nonlinear first order transient systems, solved using this method, we consistently observed that this method often yields non-physical instability in the time derivative variable of the systems. This is due to the fact that the method's control of the high frequency damping of both the primary variable and its time derivative is achieved via one parameter only (ρ_∞). In other words, the amounts of the high frequency damping for the two variables are enforced to be equal (i.e., the high frequency damping of these variables are controlled indiscriminately, and not selectively, through the one and only user control parameter ρ_∞). In our investigations using various numerical examples, we consistently observed the following: (1) often a minimal amount of numerical damping is sufficient to yield realistic numerical solutions of the primary variable; however, (2) the same amount of numerical damping often may not be sufficient to suppress the numerical oscillation in the time derivative variable resulting in oscillatory response that does not represent the true response realistically. This non-physical instability in the time derivative variable, as a consequence, can lead to physically incorrect dynamics of the system especially for long term and nonlinear simulations. From the computational point of view, it is intuitively desirable that a numerical method yields physically representative solutions of all variables involved in the system so as to completely capture the problem physics correctly. Moreover, from a practical point of view, the time derivative variable has significant importance in experimental works [64, 65]. Therefore, the non-physical instability in the time derivative variable generated by the current state-of-the-art method places a limitation to the method. A solution to this limitation is definitely to have a more flexible control of the high frequency damping, in which the numerical damping for the two variables can be enforced selectively (i.e., the analyst can select to impose a certain amount of numerical damping for the primary variable, and impose another amount of numerical damping for the time derivative variable, and additionally the two amounts of numerical damping can be the same or different), according to the needs of these variables so as to obtain physically representative solutions for both variables, i.e., solutions that correctly capture the dynamics of the problem.

Recently, we have developed and described in [66] a single step single solve time integration framework that permits such a flexible control of the high frequency damping in addition to other desirable features including second order time accuracy, unconditional stability, and zero-order overshoot behaviour for the time integration of first order transient systems, termed as *GS4-1 Framework*.[6] Such a family of time integration algorithms was developed by utilizing in a consistent manner the design procedure previously introduced for second order dynamic systems, i.e., the Algorithms by Design procedure [16, 27, 30]. Applications of the newly developed GS4-1 framework to various types of numerical examples pertaining to first order transient systems have also been described in [66–68] to demonstrate the ability and advantage of the new framework. The key step in the design procedure of the present development is the incorporation of a spurious root ρ_∞^s, in addition to the principal root ρ_∞ that appears alone in the current state-of-the-art method. For the first order transient system, the principal root of an algorithm is defined in general as the absolute maximum value of the eigenvalues of the algorithm amplification matrix. Meanwhile, a spurious root of the algorithm is defined as the absolute minimum value of the eigenvalues of such a matrix. In particular, the two parameters (ρ_∞ and ρ_∞^s) introduced in the design procedure of the GS4-1 framework represent the principal and spurious roots at high frequency. These two parameters serve as the amplification factor for the decoupled primary and time derivative variables, respectively, in the system eigen space, which numerically represent the controllable numerical dissipative property of these two variables in high frequency limit. This key design step is crucial as it allows for a selective and more flexible control of the high frequency damping for both the primary variable and its time derivative, respectively; a new feature, not available in existing methods to-date, termed hereby as the *"Selective Control Feature"*.

The significance of this new feature is for a successful simultaneous elimination of the numerical oscillation associated with the primary variable and its time derivative, yielding physically representative solutions that correctly capture the physics and

[6]Note that 'GS4-1' stands for Generalized Single Step Single Solve for 1^{st} order system, hence the consistency with the existing Generalized Single Step Single Solve for 2^{nd} order system (GS4-2)

dynamics of the problem. Such a design thereby yields a two-parameter (ρ_∞ and ρ_∞^s) family of time integration algorithms with a more flexible user control of the high frequency damping for the two variables, respectively. By allowing the two parameters to be equal (i.e, $\rho_\infty = \rho_\infty^s$), the amount of the high frequency damping for the two variables is hence equal (i.e., the high frequency damping is controlled indiscriminately, and not selectively) and the framework naturally recovers the current state-of-the-art method [62]; hence, the resulting algorithm is herein referred to as the *GS4-1 framework without the selective control feature* or *the case without the selective control feature*. We illustrate throughout this thesis how such a limited control results in non-physical instability in the time derivative variable due to insufficient damping imposed on this variable. This is consistent with our findings on the key restriction in the current state-of-the-art method (i.e., the one-parameter method) as previously mentioned; that is, the instability in the time derivative variable generated by this method is due to the restriction on the control of the high frequency damping. This restriction, as we discovered from the theoretical developments of our new GS4-1 framework, is implicitly imposed by this existing method through the algorithmic condition $\rho_\infty = \rho_\infty^s$ (i.e., without the selective control feature).

To overcome the drawback due to such a key restriction in the current state-of-the-art method, we allow the new GS4-1 framework to have a more flexible control of the high frequency damping by imposing different amounts of numerical dissipation in the two variables through the algorithmic condition $\rho_\infty \neq \rho_\infty^s$, which is the new selective control feature; hence, the resulting algorithm is referred herein as the *GS4-1 framework with the selective control feature* or *the case with the selective control feature*. Such a feature, which is new and not available in existing methods to-date, allows the simultaneous elimination of the numerical oscillation associated with the two variables to yield physically representative solutions that correctly capture the dynamics of the problem. In general, both cases (with and without the selective control feature) are contained within the newly developed GS4-1 framework described in this thesis, which is designed for first order transient systems and encompasses a family of algorithms with desirable features including second order accuracy, unconditional stability, zero-order overshoot, and controllable numerical

dissipation. In this regard, the selective control feature serves as a significant added value to the newly developed framework in comparison to existing methods in which such a feature is not available. We demonstrate throughout this thesis the ability and advantage of the GS4-1 framework via such a feature in comparison to the current state-of-the-art method using various types of numerical examples pertaining to first order transient systems. The numerical results demonstrate the roles played by this new feature in improving the numerical solutions of, not only the time derivative variable, but also the primary variable, in comparison to the existing methods without such a feature. Not only that this new feature enables the attainment of physically representative solutions of these variables, it also provides the necessary avenue to achieve this without imposing over-dissipation, which may yield numerical solutions that, although are stable, deviate from the actual dynamics of the problem as demonstrated in this thesis. Therefore, the significances of this new feature and the abilities of the newly developed framework in comparison to existing methods are successfully demonstrated. It is worth mentioning that for the purpose of demonstrating the ability and advantage of the new method in comparison to the current state-of-the-art, comparisons with softwares such as COMSOL are not considered in the thesis. This is due to the fact that COMSOL uses the one-parameter method [62] which can be recovered by the newly developed framework with choice of $\rho_\infty = \rho_\infty^s$ (i.e., without selective control feature). Since we successfully demonstrate throughout the thesis how the new method with the selective control outperforms the one-parameter method using the various numerical examples considered, comparisons with softwares such as COMSOL are already successfully represented by these examples. Additionally, comparisons with Euler Backward and Galerkin methods are not considered as these methods are only first order accurate, hence the clear disadvantage as compared to the algorithms in proposed framework which are all second order accurate. Meanwhile, algorithms pertaining to other classes of time integration method such as Runge-Kutta type methods are also not included in this thesis as these algorithms do not fall under the class of single step single solve algorithms, which is the scope of the present research.

More interestingly, the new GS4-1 framework can be naturally recovered from the existing GS4-2 framework by properly assigning the parameters and matrices, as

described in this thesis, while preserving the actual governing equations, physical interpretation and criteria of the parameters in both frameworks. This subsequently allows for the formalism of the GS4-2 framework as an isochronous integration framework that can be used to solve both first and second order dynamic systems without having to resort to the individual frameworks, hence the practicality in computational and implementation aspects. Although this may seem to highlight the significant contribution of the GS4-2 framework alone, it is important to note that the discovery of it being an isochronous integration framework had stemmed from the precise understanding of both the individual GS4-1 and GS4-2 frameworks and, in particular, of how the two frameworks are related to each other. This requires, in the first place, the successful designs and developments of the two individual frameworks. Therefore, the significance of the development of the GS4-1 framework, as presented in this thesis, remains. We now proceed to present the theoretical design and development of this new framework in the next chapter.

Chapter 3

Design and Development of the New GS4-1 Time Integration Framework

In this chapter, we present the theoretical design and development of the new GS4-1 time integration framework [66] which encompasses a family of algorithms with desirable features including: (1) second order accuracy, (2) unconditional stability, (3) zero-order overshoot behaviour, and (4) additionally possessing controllable numerical dissipation with the new selective control feature, for the time integration of first order transient system. The new framework was developed by utilizing in a consistent manner the design procedure previously introduced and successfully implemented for the second order dynamic systems, namely the Algorithms by Design procedure.

We recall from Chapter 1, that the research presented in this thesis is primarily focused on the issues and developments related to the time discretization part of the computational procedures involved in solving engineering dynamic systems. The spatial discretization part, on the other hand, is not the focus of this thesis and well-established spatial discretization methods such as the Galerkin and Streamline Upwind Petrov Galerkin (SUPG) Finite Element Methods are readily employed. Therefore, in the theoretical formulation and developments of the new GS4-1 time

integration framework to be described in detailed next, we assumed that the relevant partial differential equation governing the system has been discretized in space (i.e., semi-discretized) to yield a system of ordinary differential equations that is first order in time. How the system can be dicretized in time using the newly developed GS4-1 framework is the core motivation of the thesis, and the design and development of this new time integration framework is described next. For this purpose, we next review the Algorithms by Design procedure and then proceed to describe how the procedure was implemented for developing the new GS4-1 framework.

3.1 Review of Algorithms by Design

Algorithms by Design is a relatively new concept for the design of computational algorithms based on the unified theory presented in Ref [30] which allows one to tailor an algorithm according to pre-determined desirable algorithmic attributes, termed *wish list*. In general, the notion of Algorithms by Design is formalized by the following three steps:

1. *a priori* identify a wish list of desirable algorithmic attributes for the application at hand,
2. formulate a generalized representation of the algorithmic structure with free parameters, and
3. impose the wish list to determine the free parameters of the generalized representation such that the resulting algorithms possess the desired attributes.

Utilizing the notion of Algorithms by Design as outlined above, we first decide upon the wish list of the algorithmic properties (i.e., step (1) in the Algorithms by Design procedure). The measures for the relevant algorithmic properties for first order transient systems are accuracy, stability, numerical dissipation, and overshoot behaviour. Therefore, our desirable algorithmic attributes (wish list) are as follows: second-order accuracy, unconditional stability, zero-order overshoot behaviour, and controllable numerical dissipation with the new selective control feature. With the desirable algorithmic properties in mind, one can design the algorithms that satisfy the desirable properties by imposing the conditions for the desirable properties one

by one. We shall defer this (i.e., step (3) in the Algorithms by Design procedure) to the next section. In this section, we first present how the generalized algorithmic structure of the GS4-1 framework (i.e., with free parameters) is formulated (i.e., step (2) in the Algorithms by Design procedure), following our previous approach in developing the existing GS4-2 framework [16], which is an important step prior to imposing the conditions for the desirable properties according to the Algorithms by Design procedure. For this purpose, the existing GS4-2 framework originally developed for solving structural dynamic problems (second order dynamic system) is advanced here and re-formulated for use in first order transient applications (i.e. from a second order dynamic system to a first order dynamic system) leading to the design and developments of the new GS4-1 framework.

3.2 Theoretical Formulation of the Generalized Algorithmic Structure of the GS4-1 Framework

Consider a semi-discretized (i.e., upon the spatial discretization) system of ordinary differential equations describing a first order transient system in a general form,

$$\mathbf{M}\dot{\boldsymbol{\phi}}(t) + \mathbf{K}\boldsymbol{\phi}(t) = \mathbf{F}(t) \tag{3.2.1}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix that could be due to convection, diffusion, and surface reaction, and $\mathbf{F}$ is the force vector that could be due to the Neumann boundary conditions and sink/source terms. The formulations of these matrices vary for different types of dynamic problems, and therefore only a general representation is used in equation (3.2.1). However, specific formulations of these matrices are given in subsequent chapters where we discuss the application of the newly developed GS4-1 framework to various types of numerical examples. Additionally, in equation (3.2.1), $\boldsymbol{\phi}(t)$ and $\dot{\boldsymbol{\phi}}(t)$ represent the primary variable and its time derivative at time t that are involved in the problem (such as temperature for heat transfer problems, particle concentration in flow transport problems, and velocity in fluid dynamic problems).

Given the solutions at t_n time level (i.e., $\boldsymbol{\phi}_n$ and $\dot{\boldsymbol{\phi}}_n$), we wish to find the solutions

at t_{n+1} time level (i.e., $\boldsymbol{\phi}_{n+1}$ and $\dot{\boldsymbol{\phi}}_{n+1}$) where $t_{n+1} - t_n = \Delta t$ the time step size. For this purpose, we represent equation (3.2.1) using the classical time weighted residual approach [69] employing arbitrary weighted time field W as follows

$$\int_0^{\Delta t} W[\mathbf{M}\dot{\boldsymbol{\phi}}(t) + \mathbf{K}\boldsymbol{\phi}(t) - \mathbf{F}(t)]dt = 0 \tag{3.2.2}$$

where the weighted time field, W in equation (3.2.2) is assumed to be a degenerated scalar polynomial function of the form (see Reference [29] for details)

$$W = 1 + w_1\Gamma + w_2\Gamma^2 \tag{3.2.3}$$

where $\Gamma = \frac{t}{\Delta t}$ and $t \in [0, \Delta t]$. Meanwhile, the terms w_1 and w_2 in equation (3.2.3) are intermediate parameters to be determined later[1] after the derivation of the new framework.

We next approximate the variables $\dot{\boldsymbol{\phi}}(t)$ and $\boldsymbol{\phi}(t)$ in equation (3.2.2) using asymptotic series expansion, whereas the load term, $\mathbf{F}(t)$ is approximated using Taylor series expansion

$$\dot{\boldsymbol{\phi}}(t) = \dot{\boldsymbol{\phi}}_n + \Lambda_6\frac{\Delta\dot{\boldsymbol{\phi}}}{\Delta t}t \tag{3.2.4}$$

$$\boldsymbol{\phi}(t) = \boldsymbol{\phi}_n + \Lambda_4\dot{\boldsymbol{\phi}}_n t + \Lambda_5\frac{\Delta\dot{\boldsymbol{\phi}}}{\Delta t}t^2 \tag{3.2.5}$$

$$\mathbf{F}(t) = \mathbf{F}_n + \frac{\mathbf{F}_{n+1} - \mathbf{F}_n}{\Delta t}t \tag{3.2.6}$$

where

$$\Delta\dot{\boldsymbol{\phi}} = \dot{\boldsymbol{\phi}}_{n+1} - \dot{\boldsymbol{\phi}}_n \tag{3.2.7}$$

At this point, the newly introduced parameters Λ_4, Λ_5, and Λ_6 are also intermediate parameters. The notations and labelling of these parameters are chosen as such to ensure consistency of the new GS4-1 framework with respect to the existing GS4-2 framework allowing for the development of an isochronous integration framework

[1] see equation (3.3.86)

that can be used to solve both first and second order dynamic systems without having to resort to the individual framework (this will become more apparent in Chapter 7 where we present such a development).

Substituting the approximations, equations (3.2.4) to (3.2.6) into equation (3.2.2) and defining[2] for convenience

$$W_i = \frac{\int_0^{\Delta t} W(\frac{t}{\Delta t})^i \, dt}{\int_0^{\Delta t} W \, dt} \tag{3.2.8}$$

yields the following generalized algorithmic structure of the GS4-1 framework (in terms of free parameters, i.e, prior to imposing the wish list)

$$\mathbf{M}(\dot{\boldsymbol{\phi}}_n + \Lambda_6 W_1 \Delta\dot{\boldsymbol{\phi}}) + \mathbf{K}(\boldsymbol{\phi}_n + \Lambda_4 W_1 \Delta t \dot{\boldsymbol{\phi}}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\boldsymbol{\phi}}) = (1 - W_1)\mathbf{F}_n + W_1 \mathbf{F}_{n+1} \tag{3.2.9}$$

Therefore the semi-discretized dynamic system can now be represented in a temporally discrete form as follows

$$\mathbf{M}\widetilde{\dot{\boldsymbol{\phi}}} + \mathbf{K}\widetilde{\boldsymbol{\phi}} = \widetilde{\mathbf{F}} \tag{3.2.10}$$

where

$$\widetilde{\dot{\boldsymbol{\phi}}} = \dot{\boldsymbol{\phi}}_n + \Lambda_6 W_1 \Delta\dot{\boldsymbol{\phi}} \tag{3.2.11}$$

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_n + \Lambda_4 W_1 \Delta t \dot{\boldsymbol{\phi}}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\boldsymbol{\phi}} \tag{3.2.12}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{3.2.13}$$

The associated design for the updates of the primary variable and its time derivative at the end of each time level (i.e., $\boldsymbol{\phi}_{n+1}$ and $\dot{\boldsymbol{\phi}}_{n+1}$) are chosen, from adapting those

[2] where the subscript i on the left hand side of equation (3.2.8) is a subscript assignment to the weighted time field W according to the power of the $\frac{t}{\Delta t}$ terms contained in the approximations (equations (3.2.4) to (3.2.6))

for the second order dynamic system [16], as

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \lambda_4 \dot{\boldsymbol{\phi}}_n \Delta t + \lambda_5 \Delta \dot{\boldsymbol{\phi}} \Delta t \tag{3.2.14}$$

$$\dot{\boldsymbol{\phi}}_{n+1} = \dot{\boldsymbol{\phi}}_n + \Delta \dot{\boldsymbol{\phi}} \tag{3.2.15}$$

At this point, the terms $\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5 are free parameters that will be defined later in Section 3.3 by imposing the conditions for achieving the desired algorithmic attributes.

Algorithm 1

Generalized Algorithmic Structure of the GS4-1 Framework (in terms of free parameters, i.e., prior to imposing the wish list)

Consider a semi-discretized first order transient system in a general form of representation as follows,

$$\mathbf{M}\dot{\boldsymbol{\phi}}(\mathrm{t}) + \mathbf{K}\boldsymbol{\phi}(\mathrm{t}) = \mathbf{F}(\mathrm{t}) \tag{3.2.16}$$

Given $\boldsymbol{\phi}_n$ and $\dot{\boldsymbol{\phi}}_n$, we can find the solutions at the next time level $\boldsymbol{\phi}_{n+1}$ and $\dot{\boldsymbol{\phi}}_{n+1}$ by first solving for $\Delta\dot{\boldsymbol{\phi}}$ from

$$\begin{aligned}(\Lambda_6 W_1 \mathbf{M} + \Lambda_5 W_2 \Delta t \mathbf{K})\Delta\dot{\boldsymbol{\phi}} = &- \mathbf{M}\dot{\boldsymbol{\phi}}_n - \mathbf{K}(\boldsymbol{\phi}_n + \Lambda_4 W_1 \Delta t \dot{\boldsymbol{\phi}}_n) \\ &+ \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n)\end{aligned} \tag{3.2.17}$$

followed by updating the variables at the end of each time level according to

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \lambda_4 \dot{\boldsymbol{\phi}}_n \Delta t + \lambda_5 \Delta \dot{\boldsymbol{\phi}} \Delta t \tag{3.2.18}$$

$$\dot{\boldsymbol{\phi}}_{n+1} = \dot{\boldsymbol{\phi}}_n + \Delta \dot{\boldsymbol{\phi}} \tag{3.2.19}$$

Remark 3.2.1

1. *Algorithm 1 is the structure of the GS4-1 framework in generalized form with free parameters prior to imposing the conditions for achieving the desired algorithmic attributes.*

2. *Utilizing the concept of Algorithms by Design, a wish list of the desirable algorithmic properties is decided upon and imposed to the generalized framework to determine the free parameters.*

3. *In Algorithm 1, there is a total of 6 free parameters:* $\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, *and* λ_5. *These free parameters are related to* Λ_i *and* λ_i *contained in the approximations and updates of the variables, in combination with* W_i *associated with the weighted time field. Consequently, they entail unique relations which serve as the discrete numerically assigned [DNA] algorithmic markers which are the algorithm signature.*

Relevant details follow next.

3.3 Tailoring the Generalized Algorithmic Structure of the GS4-1 Framework Towards Achieving Desirable Algorithmic Attributes

In this section, we proceed by demonstrating how the desirable algorithmic properties (wish list) can be achieved by imposing the appropriate conditions for these desirable attributes to the generalized algorithmic structure of the GS4-1 framework (*Algorithm* 1). For this purpose, it is worth highlighting that the algorithmic properties for multi-degree of freedom (MDOF) problems, such as those represented by equation (3.2.16), are equivalent to that of single-degree of freedom (SDOF) model problems [16, 62]. This is due to the satisfaction of the mode superposition technique which states that the response of linear systems governing MDOF problems is the sum of the responses of each individual mode (i.e., SDOF). That is, equation (3.2.16) can be uncoupled into many SDOF problems [62], each of which retains the same algorithmic properties as those for equation (3.2.16). Therefore, for ease of explanation of how the conditions for the desirable algorithmic properties can be imposed on the generalized framework (*Algorithm* 1), we consider in this section the SDOF model problem of equation (3.2.16), which can be expressed in a general representation as follows

$$\dot{\phi} + \lambda\phi = f \tag{3.3.20}$$

where λ is the eigenvalue associated with the chosen mode [62]. According to *Algorithm* 1, this SDOF model problem (equation (3.3.20)) can be solved at each time

level by first solving for $\Delta\dot{\phi}$ from

$$(\Lambda_6 W_1 + \Lambda_5 W_2 \Delta t\lambda)\Delta\dot{\phi} = -\dot{\phi}_n - \lambda(\phi_n + \Lambda_4 W_1 \Delta t\dot{\phi}_n) + f_n + W_1(f_{n+1} - f_n) \tag{3.3.21}$$

followed by updating the variables at the end of each time level according to

$$\phi_{n+1} = \phi_n + \lambda_4\dot{\phi}_n\Delta t + \lambda_5\Delta\dot{\phi}\Delta t \tag{3.3.22}$$

$$\dot{\phi}_{n+1} = \dot{\phi}_n + \Delta\dot{\phi} \tag{3.3.23}$$

To impose the conditions for the desirable attributes, we first represent the temporally discrete equation for the model problem in the following form [70]

$$\mathbf{d}_{n+1} = \mathbf{A}\mathbf{d}_n + \mathbf{L}_n \tag{3.3.24}$$

where

$$\mathbf{d}_n = (\phi_n \ \ \Delta t\dot{\phi}_n)^T \tag{3.3.25}$$

$$\mathbf{d}_{n+1} = (\phi_{n+1} \ \ \Delta t\dot{\phi}_{n+1})^T \tag{3.3.26}$$

and $\mathbf{A}$ is the amplification matrix which will be greatly used in the subsequent design steps. From equations (3.3.21) to (3.3.23), we can find that $\mathbf{A}$, as defined in equation (3.3.24), is given as

$$\mathbf{A} = \begin{bmatrix} 1 + \lambda_5 A_{21} & \lambda_4 - \lambda_5(1 - A_{22}) \\ A_{21} & A_{22} \end{bmatrix} \tag{3.3.27}$$

where

$$A_{21} = -\frac{\Omega}{D} \tag{3.3.28}$$

$$A_{22} = 1 - \frac{1}{D}(1 + \Lambda_4 W_1 \Lambda) \tag{3.3.29}$$

$$D = \Lambda_6 W_1 + \Lambda_5 W_2 \Omega \tag{3.3.30}$$

$$\Omega = \lambda \Delta t \tag{3.3.31}$$

Meanwhile, $\mathbf{L}_n$ is the corresponding load vector. From equations (3.3.21) to (3.3.23), we can find that $\mathbf{L}_n$, as defined in equation (3.3.24), is given as

$$\mathbf{L}_n = \frac{1}{D} \left\{ \begin{array}{c} \lambda_5 \Delta t[(1 - W_1) f_n + W_1 f_{n+1}] \\ \\ \Delta t[(1 - W_1) f_n + W_1 f_{n+1}] \end{array} \right\} \tag{3.3.32}$$

We next describe how the conditions for achieving the desirable algorithmic properties (wish list) can be imposed one by one on the generalized algorithmic structure of the GS4-1 framework (*Algorithm* 1).

3.3.1 Second Order Accuracy

In solving dynamic problems numerically/computationally, an important factor to consider in selecting the time integration method to be used is the method's order of accuracy. This is due to the fact that in deriving these methods, approximations are introduced and, as a consequence, errors are generated. These errors, which are the difference between the approximations and the exact analytical solutions, depend on the time step size Δt. As smaller value of Δt is used, the errors generated by the method should also decrease[3]. The rate at which the temporal error decreases with Δt is the order of accuracy of the method. If the errors decreases linearly with Δt, the method is said to be first order accurate. Meanwhile, a second order accuracy

[3]theoretically, as Δt approaches zero the approximations should approach the real solutions and therefore the errors approach zero. Practically, however, the roundoff error of finite arithmetic increases as Δt approaches zero. This means that the numerical error never goes to zero but reaches a minimum value past which it actually begins to increase as Δt approaches zero due to the prominence of the roundoff error

indicates that the errors generated by the method decreases quadratically with Δt. Clearly, a higher-order method is desirable as the error reduces more rapidly with smaller Δt.

However, as explained in [16], the algorithmic properties of the single step single solve classification of algorithms are bounded by the Dahlquist theorem [71]. According to the Dahlquist theorem, if an algorithm is unconditionally stable, it can be at most second order accurate. Since unconditional stability is a desirable algorithmic attribute that we wish to impose on *Algorithm* 1 to ensure bounded solutions for any time step sizes (Δt), the algorithms developed in this work will have to be at most second order accurate due to the limitation of the Dahlquist barrier theorem. Despite such a limitation, it is worth mentioning that second order accuracy is a desirable algorithmic attribute, in comparison to the first order accuracy such as those algorithms of the classical Trapezoidal family (with the exception of the Crank-Nicolson method which is second order accurate).

We next describe how the second order accuracy can be imposed on *Algorithm* 1. In this regard, an expression of interest for the accuracy of an algorithm is the local truncation error $\tau(t)$. It represents the error in the numerical solution produced by the algorithm for a single time step (i.e., from t_n to t_{n+1} where $t_{n+1} - t_n = \Delta t$). If the numerical solutions $\mathbf{d}_n$ and $\mathbf{d}_{n+1}$ in equation (3.3.24) are replaced by the corresponding exact values ($\mathbf{d}(t_n)$ and $\mathbf{d}(t_{n+1})$), we obtain the following expression [70]

$$\mathbf{d}(t_{n+1}) - \mathbf{A}\mathbf{d}(t_n) - \mathbf{L}_n = \tau(t_n)\Delta t \tag{3.3.33}$$

where $\tau(t_n)$ is called the local truncation error. Therefore, the local truncation error is defined in general as follows

$$\tau(t_n) = \frac{1}{\Delta t}[\mathbf{d}(t_{n+1}) - \mathbf{A}\mathbf{d}(t_n) - \mathbf{L}_n] \tag{3.3.34}$$

An algorithm is p^{th} order accurate if p is the largest integer such that, for all $t \in [0, T]$

(where T is the end time of the transient duration) and $\Delta t > 0$,

$$|\tau(t)| \leq c\Delta t^p = O(\Delta t^p) \tag{3.3.35}$$

where c is a constant independent of Δt, and $p > 0$; p is called the order of accuracy or rate of convergence [70].

In accordance with equation (3.3.34), we now derive expression for $\tau(t_n)$ specific for first order transient systems. Recall from equation (3.3.24) that we have

$$\mathbf{d}_{n+1} = \mathbf{A}\mathbf{d}_n + \mathbf{L}_n \tag{3.3.36}$$

Likewise, we have

$$\mathbf{d}_n = \mathbf{A}\mathbf{d}_{n-1} + \mathbf{L}_{n-1} \tag{3.3.37}$$

Substituting equation (3.3.37) into equation (3.3.36), we have

$$\begin{aligned} \mathbf{d}_{n+1} &= \mathbf{A}(\mathbf{A}\mathbf{d}_{n-1} + \mathbf{L}_{n-1}) + \mathbf{L}_n \\ &= \mathbf{A}^2\mathbf{d}_{n-1} + \mathbf{A}\mathbf{L}_{n-1} + \mathbf{L}_n \end{aligned} \tag{3.3.38}$$

By the Cayley - Hamilton theorem [72], we have

$$\mathbf{A}^2 + A_1\mathbf{A} + A_2 = 0 \tag{3.3.39}$$

where

$$A_1 = -Tr(\mathbf{A})$$
$$A_2 = Det(\mathbf{A})$$

Substituting equation (3.3.39) into equation (3.3.38) yields

$$\begin{aligned}\mathbf{d}_{n+1} &= (-A_1\mathbf{A} - A_2)\mathbf{d}_{n-1} + \mathbf{A}\mathbf{L}_{n-1} + \mathbf{L}_n \\ &= -A_1\mathbf{A}\mathbf{d}_{n-1} - A_2\mathbf{d}_{n-1} + \mathbf{A}\mathbf{L}_{n-1} + \mathbf{L}_n \end{aligned} \tag{3.3.40}$$

Using equation (3.3.37) and rearranging, we obtain the following

$$\mathbf{d}_{n+1} + A_1\mathbf{d}_n + A_2\mathbf{d}_{n-1} = (A_1\mathbf{I} + \mathbf{A})\mathbf{L}_{n-1} + \mathbf{L}_n \tag{3.3.41}$$

where $\mathbf{I}$ is a 2×2 identity matrix. Replacing the numerical solutions $\mathbf{d}_{n+1}$, $\mathbf{d}_n$ and $\mathbf{d}_{n-1}$ by the corresponding exact values ($\mathbf{d}(t_{n+1})$, $\mathbf{d}(t_n)$ and $\mathbf{d}(t_{n-1})$) (in accordance with equation (3.3.34)) we have the following expression defining the local truncation error for the first order transient system defined by: $\dot{\phi} + \lambda\phi = f$ (i.e., equation (3.3.20))

$$\tau(t_n) = \frac{1}{\Delta t}[\phi(t_{n+1}) + A_1\phi(t_n) + A_2\phi(t_{n-1}) - F] \tag{3.3.42}$$

where F is the first row of the vector $\{(A_1\mathbf{I} + \mathbf{A})\mathbf{L}_{n-1} + \mathbf{L}_n\}$ and is obtained as

$$\begin{aligned} F = \frac{1}{D}\{&(A_1 + 1 + \lambda_5 A_{21})\lambda_5\Delta t[(1 - W_1)f(t_{n-1}) + W_1 f(t_n)] \\ &+ [\lambda_4 - \lambda_5(1 - A_{22})]\Delta t[(1 - W_1)f(t_{n-1}) + W_1 f(t_n)] \\ &+ \lambda_5\Delta t[(1 - W_1)f(t_n) + W_1 f(t_{n+1})]\} \end{aligned} \tag{3.3.43}$$

For the amplification matrix $\mathbf{A}$ given by equation (3.3.27), the resulting trace and

determinant of the matrix are

$$Tr(\mathbf{A}) = \mathbf{I} : \mathbf{A} = 2 - \frac{1}{D}\left[1 + (\lambda_5 + \Lambda_4 W_1)\Omega\right] \tag{3.3.44}$$

$$Det(\mathbf{A}) = 1 - \frac{1}{D}\left[1 + (\lambda_5 - \lambda_4 + \Lambda_4 W_1)\Omega\right] \tag{3.3.45}$$

Additionally, the terms $\phi(t_{n+1})$, $\phi(t_{n-1})$, $f(t_{n+1})$, and $f(t_{n+1})$ are expressed using Taylor series expansion at time t_n as follows

$$\begin{aligned}
\phi(t_{n+1}) &= \phi(t_n) + \dot{\phi}(t_n)\Delta t + \frac{1}{2}\ddot{\phi}(t_n)\Delta t^2 \\
\phi(t_{n-1}) &= \phi(t_n) - \dot{\phi}(t_n)\Delta t + \frac{1}{2}\ddot{\phi}(t_n)\Delta t^2 \\
f(t_{n+1}) &= f(t_n) + \dot{f}(t_n)\Delta t + \frac{1}{2}\ddot{f}(t_n)\Delta t^2 \\
f(t_{n-1}) &= f(t_n) - \dot{f}(t_n)\Delta t + \frac{1}{2}\ddot{f}(t_n)\Delta t^2
\end{aligned}$$

Doing the necessary substitutions, performing the arithmetic operations and rearranging, equation (3.3.42) becomes

$$\begin{aligned}
\tau(t_n) = {} & \frac{1}{D}[\dot{\phi}(t_n) + \lambda_4\lambda\phi(t_n) - \lambda_4 f(t_n)] \\
& + \frac{\Delta t}{D}\left\{\left(\Lambda_6 W_1 - \frac{1}{2}\right)\ddot{\phi}(t_n) + (\lambda_5 - \lambda_4 + \Lambda_4 W_1)\lambda\dot{\phi}(t_n) - [\lambda_5 - \lambda_4(1 - W_1)]\dot{f}(t_n)\right\} \\
& + \frac{\Delta t^2}{D}\left\{\ddot{\phi}(t_n)\lambda\left[\Lambda_5 W_2 - \frac{1}{2}(\lambda_5 - \lambda_4 + \Lambda_4 W_1)\right] + \ddot{f}(t_n)\left[\lambda_5 W_1 + \frac{1}{2}[\lambda_4(1 - W_1) - \lambda_5]\right]\right\}
\end{aligned} \tag{3.3.46}$$

Therefore, to achieve second order time accuracy, the following three conditions/constraints need to be imposed on *Algorithm* 1

$$\lambda_4 = 1 \tag{3.3.47}$$

$$\Lambda_6 W_1 - \frac{1}{2} = \lambda_5 - \lambda_4 + \Lambda_4 W_1 \tag{3.3.48}$$

$$\lambda_5 - \lambda_4(1 - W_1) = \lambda_5 - \lambda_4 + \Lambda_4 W_1 \tag{3.3.49}$$

By imposing the above conditions, equation (3.3.46) becomes

$$\begin{aligned}\tau(t_n) =& \frac{1}{D}\left\{[\dot{\phi}(t_n) + \lambda\phi(t_n) - f(t_n)] + \Delta t(\lambda_5 - 1 + \Lambda_4 W_1)[\ddot{\phi}(t_n) + \lambda\dot{\phi}(t_n) - \dot{f}(t_n)]\right\} \\ &+ \frac{1}{D}\left\{O(\Delta t^2) + ...\right\}\end{aligned} \tag{3.3.50}$$

An observation of equation (3.3.50) and recalling from equation (3.3.20), we can see that

$$\tau(t_n) = O(\Delta t^2) \tag{3.3.51}$$

which indicates that second order accuracy is achieved.

3.3.2 Zero-Order Overshoot Behaviour

The overshoot behaviour is a numerically induced property/attribute of a numerical algorithm at high frequency limit. At this limit, the numerical solution overshoots the exact solution [16]. For large-scale engineering problems with a wide range of natural frequency, algorithms with higher than zero-order overshoot behaviour will exhibit overshoot in the first few time steps as the value of Δt increases. This may manifest in the subsequent computations and will lead to an increase in iterations, especially in the context of large-scale practical nonlinear dynamic problems [16]. Therefore, it is desirable that the algorithms in the GS4-1 framework induces no more than a zero-order overshoot behaviour on the primary variable. This means that the overshoot on the solution of the primary variable at high frequency limit is independent of Δt. That is, the amount of overshoot does not increase although Δt is increased; hence is a desirable feature.

To impose the condition related to the overshoot behaviour, we consider the amplifi-

cation matrix in the high frequency limit, defined as follows

$$\mathbf{A}_\infty = \lim_{\Omega\to\infty} \mathbf{A} \tag{3.3.52}$$

$$= \begin{bmatrix} 1 - \frac{\lambda_5}{\Lambda_5 W_2} & \lambda_4 - \lambda_5 \frac{\Lambda_4 W_1}{\Lambda_5 W_2} \\ -\frac{1}{\Lambda_5 W_2} & 1 - \frac{\Lambda_4 W_1}{\Lambda_5 W_2} \end{bmatrix} \tag{3.3.53}$$

Denoting the initial conditions as follows,

$$\mathbf{d}_0 = \begin{pmatrix} \phi_0 \\ \Delta t \dot{\phi}_0 \end{pmatrix} \tag{3.3.54}$$

the solutions at the next time level at high frequency limit can be represented as

$$\mathbf{d}_1 = \mathbf{A}_\infty \mathbf{d}_0 \tag{3.3.55}$$

where

$$\mathbf{d}_1 = \begin{pmatrix} \phi_1 \\ \Delta t \dot{\phi}_1 \end{pmatrix} \tag{3.3.56}$$

Therefore, the solution of the primary variable at the high frequency limit is obtained as follows

$$\phi_1 = \left(1 - \frac{\lambda_5}{\Lambda_5 W_2}\right) \phi_0 + \left(\lambda_4 - \lambda_5 \frac{\Lambda_4 W_1}{\Lambda_5 W_2}\right) \Delta t \dot{\phi}_0 \tag{3.3.57}$$

From equation (3.3.57), therefore, a zero-order overshoot behaviour on the primary variable requires that

$$\lambda_4 - \lambda_5 \frac{\Lambda_4 W_1}{\Lambda_5 W_2} = 0 \tag{3.3.58}$$

Imposing the condition for second order time accuracy, equation (3.3.47), to equation (3.3.58) we have the following constraint for the zero-order overshoot behaviour

$$\lambda_5 \, \Lambda_4 W_1 = \Lambda_5 W_2 \tag{3.3.59}$$

By imposing the above constraint, equation (3.3.57) becomes

$$\phi_1 = \left(1 - \frac{\lambda_5}{\Lambda_5 W_2}\right) \phi_0 \tag{3.3.60}$$

An observation of equation (3.3.60) indicates that the overshoot on the solution of the primary variable is independent of Δt, i.e., zero-order overshoot is achieved.

3.3.3 Unconditional Stability

The stability of the time integration methods is another important criterion to look for when selecting algorithm for solving dynamic problems. This is due to the fact that the time step size Δt values that can be used are limited by the stability property of the algorithm, which can be categorized either being: (1) unconditional stability, or (2) conditional stability, depending on the eigenvalues (ξ) of the amplification matrix. The methods with the former stability property would either ensure a smooth decay ($0 \leq \xi \leq 1$) which is non-oscillatory, or fall under the oscillatory decay range ($-1 \leq \xi < 0$) for large Δt values (i.e., at the high frequency limit). In both cases, the transient responses generated by the unconditionally stable algorithms always remain bounded for any Δt. However, for methods with the latter stability property, there is a condition that if Δt is chosen to be more than a critical value, the transient response generated by the method not only may exhibit unrealistic behaviour but also may grow with time and eventually becomes unbounded ($|\xi| > 1$). For this reason, it is desirable that the algorithms in the present development are all unconditionally stable.

Stability can be assessed by looking at the eigenvalues of the amplification matrix, $\mathbf{A}$. As described above, unconditional stability will be attained so long as the modulus of each eigenvalue is less than or equal to one.

The eigenvalues of $\mathbf{A}$, $\xi_{i\,(i=1,2)}$ can be obtained from

$$Det(\mathbf{A} - \xi\mathbf{I}) = 0 \tag{3.3.61}$$

Therefore we have

$$\begin{aligned}\xi_{i\,(i=1,2)} =& \frac{1}{2}\left(2 - \frac{1}{D}[1 + (\lambda_5 + \Lambda_4 W_1)\Omega]\right) \\ & \pm \frac{1}{2}\sqrt{\frac{1}{D^2}[1 + 2(\lambda_5 + \Lambda_4 W_1)\Omega + (\lambda_5 + \Lambda_4 W_1)^2\Omega^2] - \frac{4}{D}\lambda_4\Omega}\end{aligned} \tag{3.3.62}$$

where

$$\begin{aligned}D &= \Lambda_6 W_1 + \Lambda_5 W_2 \Omega \\ \Omega &= \lambda \Delta t\end{aligned}$$

We will illustrate the stability constraints on the algorithmic parameters through the limiting values of Ω as follows:

(i) First consider the case when the time step size (Δt) is taken to be very small (i.e., low frequency limit). Regardless of the value of λ, Ω vanishes, and the eigenvalues of $\mathbf{A}$ at this limit are

$$\xi_i^0 = \lim_{\Delta t \to 0} \xi_i = \lim_{\Omega \to 0} \xi_i \tag{3.3.63}$$

$$= \frac{1}{2}\left(2 - \frac{1}{\Lambda_6 W_1} \pm \frac{1}{\Lambda_6 W_1}\right) \tag{3.3.64}$$

Therefore

$$\xi_i^0 = \lim_{\Delta t \to 0} \xi_i = \left\{ 1, 1 - \frac{1}{\Lambda_6 W_1} \right\} \tag{3.3.65}$$

from which we may deduce that unconditional stability requires

$$\left| 1 - \frac{1}{\Lambda_6 W_1} \right| \leq 1 \tag{3.3.66}$$

(ii) We next consider the limit of an infinite Δt (i.e., high frequency limit). The eigenvalues of $\mathbf{A}$ at this limit are

$$\begin{aligned} \xi_i^\infty &= \lim_{\Delta t \to \infty} \xi_i = \lim_{\Omega \to \infty} \xi_i \\ &= \frac{1}{2} \left(2 - \frac{\lambda_5 + \Lambda_4 W_1}{\Lambda_5 W_2} \pm \frac{\lambda_5 - \Lambda_4 W_1}{\Lambda_5 W_2} \right) \end{aligned} \tag{3.3.67}$$

Therefore

$$\xi_i^\infty = \lim_{\Delta t \to \infty} \xi_i = \left\{ 1 - \frac{\lambda_5}{\Lambda_5 W_2}, 1 - \frac{\Lambda_4 W_1}{\Lambda_5 W_2} \right\} \tag{3.3.68}$$

from which we may deduce that unconditional stability requires

$$\left| 1 - \frac{\lambda_5}{\Lambda_5 W_2} \right| \leq 1, \quad \left| 1 - \frac{\Lambda_4 W_1}{\Lambda_5 W_2} \right| \leq 1 \tag{3.3.69}$$

3.3.4 Controllable Numerical Dissipation with the New Selective Control Feature

Numerical dissipation is also called algorithmic dissipation or algorithmic/numerical damping, and it represents the dissipative effect induced by the algorithm at high frequency limit (hence sometimes referred to as the high frequency damping) [16]. It is an artifact that is added into the system to stabilize the algorithm in order to yield physically representative/accurate numerical solutions. Algorithms with numerical dissipation are often called the numerically dissipative algorithms. Whilst on one

hand numerical dissipation is useful to overcome the shortcomings of nondissipative schemes (which often yield numerical solutions with non-physical instabilities at large time step size), the use of it must be done in a strategic manner so as to avoid imposing over-dissipation (i.e., imposing greater amount of numerical dissipation than is needed) into the system. The importance of this delicate balancing act requires that the numerical dissipation be controlled smartly by the analyst depending on the needs of the problem of interest.

The existing one-parameter controllable numerically dissipative algorithms for first order transient systems [62], however, lack flexibility due to the restriction that the amount of high frequency damping for the primary variable and its time derivative is enforced to be equal (i.e., the high frequency damping of these variables are controlled indiscriminately, and not selectively, through the one and only user control parameter ρ_∞). From our investigations, this restriction often leads to non-physical instability in the time derivative variable. As previously mentioned, from a computational point of view, it is intuitively desirable that a numerical method yields physically representative solutions of all variables involved in the system so as to completely capture the problem physics correctly. Moreover, from a practical point of view, the time derivative variable has significant importance in some experimental works [64, 65]. Therefore, the instability in the time derivative variable generated by this current state-of-the-art method places a limitation to the method due to the restriction on the control of the high frequency damping as discussed above.

To overcome this drawback, we allow for a more flexible control of the high frequency damping, in which the numerical damping for the primary variable and its time derivative can be enforced selectively, according to the needs of these variables so as to obtain physically representative solutions of both variables. To achieve this, we first recall that *Algorithm* 1 corresponding to the SDOF model equation can be written in the form of equation (3.3.24), and then the eigenvalues of the algorithm amplification matrix $\mathbf{A}$ can be solved from

$$Det(\mathbf{A} - \xi\mathbf{I})\mathbf{X} = 0 \tag{3.3.70}$$

where $\mathbf{X}$ is the unified right eigenvector matrix, and the eigenvalues are given as ξ_1 and ξ_2 by equation (3.3.62). Mapping the primary variable and its time derivative ($\boldsymbol{\phi}$ and $\dot{\boldsymbol{\phi}}$) onto the eigenspace yields,

$$\begin{pmatrix} \bar{\boldsymbol{\phi}} \\ \bar{\dot{\boldsymbol{\phi}}} \end{pmatrix} = \mathbf{X}^{-1} \begin{pmatrix} \boldsymbol{\phi} \\ \dot{\boldsymbol{\phi}} \end{pmatrix} \tag{3.3.71}$$

from which we obtain the following

$$\bar{\boldsymbol{\phi}}_n = \xi_1^n \bar{\boldsymbol{\phi}}_0 \tag{3.3.72}$$

$$\bar{\dot{\boldsymbol{\phi}}}_n = \xi_2^n \bar{\dot{\boldsymbol{\phi}}}_0 \tag{3.3.73}$$

It is to be noted that in the above expressions, we do not consider the coupling between the two variables as we have considered the mapping of the variables onto the eigenspace. However, in practice, such coupling is very minor as demonstrated by the results of the considered numerical examples. By referring to equations (3.3.72) and (3.3.73), therefore, we can see that the eigenvalues, ξ_1 and ξ_2, serve as the amplification factor for the decoupled primary variable and its time derivative, $\boldsymbol{\phi}$ and $\dot{\boldsymbol{\phi}}$, respectively, in the system eigenspace. At high frequency limit ($n \to \infty$), the dissipative effects induced by the algorithm on these variables are related to the eigenvalues in this limit (ξ_1^∞ and ξ_2^∞) given by equation (3.3.68) [16]. Consequently, to achieve a selective and more flexible control on the high frequency damping of these variables (i.e., controllable numerical dissipation with the new selective control feature), we recall equation (3.3.68) and now introduce two user-control parameters ρ_∞ and ρ_∞^s such that

$$\xi_1^\infty = 1 - \frac{\lambda_5}{\Lambda_5 W_2} = -\rho_\infty \tag{3.3.74}$$

$$\xi_2^\infty = 1 - \frac{\Lambda_4 W_1}{\Lambda_5 W_2} = -\rho_\infty^s \tag{3.3.75}$$

It is important to note from the above equations that the two newly introduced

parameters (ρ_∞ and ρ_∞^s) are related to the eigenvalues of the amplification matrix at high frequency (ξ_1^∞ and ξ_2^∞), respectively, and therefore directly influence the amount of high frequency damping for the primary variable and its time derivative ($\boldsymbol{\phi}$ and $\dot{\boldsymbol{\phi}}$), respectively. By incorporating these two parameters (i.e., ρ_∞ and ρ_∞^s) into the design of the present framework as shown above, the analyst has a selective and more flexible control on the amount of high frequency damping he or she wishes to impose on the two variables depending on the selection for the values of these parameters which are user-control. This selection depends on the needs of the problem at hand and additionally has to satisfy certain relation. This is discussed next.

Recall that the absolute maximum value of the eigenvalues of the algorithm amplification matrix defines the so-called principal root of the algorithm. Meanwhile, the absolute minimum value of the eigenvalues defines the so-called spurious root of the algorithm. Since $\xi_1^\infty \leq \xi_2^\infty$ (see equation (3.3.68)), it follows from equations (3.3.74) and (3.3.75) that $\rho_\infty \geq \rho_\infty^s$ (due to the effect of the negative sign incorporated into these equations). Consequently, we have the following relation between the eigenvalues and the two user-control parameters

$$\rho_\infty = |\xi_1^\infty| \geq \rho_\infty^s = |\xi_2^\infty| \tag{3.3.76}$$

Therefore, ρ_∞ represents the principal root of the algorithm at high frequency and is associated with the numerical dissipation of the primary variable in this limit. Meanwhile, ρ_∞^s represents the spurious root of the algorithm at high frequency and is related to the numerical dissipation of the time derivative variable in this limit. For convenience, these parameters (ρ_∞ and ρ_∞^s) are referred to in this thesis as the principal and spurious roots of the algorithm, respectively. Due to the definitions of these roots, accordingly, these parameters must satisfy the condition that $\rho_\infty \geq \rho_\infty^s$ as pointed out above. Additionally, recall from Section 3.3.3 that the modulus of the eigenvalues must be less than or equal to one in order for the algorithms to achieve unconditional stability. In particular, at the high frequency limit (i.e., for large Δt values) unconditional stability requires that the eigenvalues fall under the range $-1 < \xi_i^\infty < 0$ (see Section 3.3.3). Accordingly, the values of the two parameters (ρ_∞ and ρ_∞^s) introduced above also have to range between 1 and 0 (note that the negative sign

appearing in the range of ξ_i^∞ is already incorporated in equations (3.3.74) and (3.3.75), and therefore the two parameters (ρ_∞ and ρ_∞^s) only take the absolute value between 1 and 0). Putting together, we have the following relation of the two parameters needed to be satisfied when selecting their values:

$$0 \leq \rho_\infty^s \leq \rho_\infty \leq 1 \tag{3.3.77}$$

From equations (3.3.72) and (3.3.73), and recalling the relation between the eigenvalues and the two user-control parameters as discussed above (see equation (3.3.76)), we can easily see that for each parameter, a value of unity means that no numerical dissipation is imposed on the respective variable (i.e., nondissipative). On the other hand, a value of zero results in imposing maximum numerical dissipation on the respective variable. Additionally, when different values are selected for the two parameters satisfying the relation in equation (3.3.77) (i.e., when selecting $\rho_\infty^s < \rho_\infty$), the resulting algorithm is said to impose different amount of numerical dissipation on the two variables, in which case the selective control feature is enabled. On the other hand, when the values of the two parameters are selected to be equal (i.e., $\rho_\infty^s = \rho_\infty$), the amount of the numerical dissipation on the two variables is hence equal (i.e., the numerical dissipation property of the algorithm is controlled indiscriminately and not selectively) and the resulting algorithm naturally recovers the existing one-parameter controllable numerically dissipative method described in [62]. Furthermore, when $\rho_\infty^s = \rho_\infty = 1$, the nondissipative Crank-Nicolson method is recovered, while the Gear's method [63] is recovered with selection of $\rho_\infty^s = \rho_\infty = 0$ [62]. The recovery of these existing methods from the present development is demonstrated in Section 4.2. In other words, the two parameters, ρ_∞ and ρ_∞^s, uniquely define particular algorithms within the GS4-1 framework (therefore the algorithm in the GS4-1 framework is often denoted, for example in illustrative figures and tabulated tables, as: GS4-1(ρ_∞,ρ_∞^s)).

Remark 3.3.1

- *ρ_∞ represents the principal root of the algorithm at high frequency limit and is associated with the numerical dissipation property of the primary variable ($\boldsymbol{\phi}$). Its value may range from 0 to 1; $\rho_\infty = 1$ means that no numerical dissipation (nondissipative/zero damping) is imposed on $\boldsymbol{\phi}$ while $\rho_\infty = 0$ means that*

maximum numerical dissipation is imposed on $\boldsymbol{\phi}$.

- *ρ_∞^s represents the spurious root of the algorithm at high frequency limit and is associated with the numerical dissipation property of the time derivative variable ($\dot{\boldsymbol{\phi}}$). Its value may range from 0 to 1; $\rho_\infty^s = 1$ means that no numerical dissipation (nondissipative/zero damping) is imposed on $\dot{\boldsymbol{\phi}}$ while $\rho_\infty^s = 0$ means that maximum numerical dissipation is imposed on $\dot{\boldsymbol{\phi}}$.*
- *Additionally, the selection for the values of these parameters must satisfy the following relation:* $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$

From this point, all of the free parameters ($\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5) appearing in *Algorithm* 1 will be expressed in terms of these two new parameters (ρ_∞ and ρ_∞^s)[4]. This is done as follows.

Substituting equation (3.3.59) into equation (3.3.74) yields

$$\Lambda_4 W_1 = \frac{1}{1+\rho_\infty} \tag{3.3.78}$$

and substituting equation (3.3.59) into equation (3.3.75) yields

$$\lambda_5 = \frac{1}{1+\rho_\infty^s} \tag{3.3.79}$$

Using equations (3.3.78) and (3.3.79), we can express all the other free parameters in terms of ρ_∞ and ρ_∞^s as follows. From equation (3.3.48), we get

$$\Lambda_6 W_1 = \frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)(1+\rho_\infty^s)} \tag{3.3.80}$$

Meanwhile, from equation (3.3.49), we obtain

$$W_1 = \frac{1}{1+\rho_\infty} \tag{3.3.81}$$

[4]except for λ_4 since we have already figured out that $\lambda_4 = 1$ (see equation (3.3.47))

and finally, from equation (3.3.59) we get

$$\Lambda_5 W_2 = \frac{1}{(1+\rho_\infty)(1+\rho_\infty^s)} \tag{3.3.82}$$

By imposing all of the conditions described above, the free parameters ($\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5) appearing in *Algorithm* 1, which is the generalized algorithmic structure of the GS4-1 framework prior to imposing the wish list, are now expressed in terms of the two parameters of the framework, ρ_∞ and ρ_∞^s, as follows

$$\begin{aligned}
\Lambda_6 W_1 &= \frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)(1+\rho_\infty^s)} \\
\Lambda_5 W_2 &= \frac{1}{(1+\rho_\infty)(1+\rho_\infty^s)} \\
\Lambda_4 W_1 &= \frac{1}{1+\rho_\infty} \\
W_1 &= \frac{1}{1+\rho_\infty} \\
\lambda_4 &= 1 \\
\lambda_5 &= \frac{1}{1+\rho_\infty^s}
\end{aligned} \tag{3.3.83}$$

At this point, it is worth clarifying the following:

- In the above design of the new GS4-1 time integration framework, the chosen algorithmic attributes, namely second-order time accuracy, unconditional stability, zero-order overshoot, and controllable numerical dissipation with the new selective control feature, are all desirable attributes to look for when selecting time integration algorithms to solve dynamic problems as described above. In particular, the latter is the new and original contribution of the present development not available in existing methods for first order transient systems to-date. This new feature enables the attainment of physically representative solutions of all variables necessary to completely capture the dynamics of the problem correctly. Hence, it serves as a significant added value to the new framework in addition to all the other desirable attributes. Following this, most of the

discussion in this thesis is focused on the demonstration of the effect and ability of this new feature. However, for completeness, we also provide demonstration of the other algorithmic properties although this is discussed only briefly for proof and illustration purposes.

- The order at which the conditions associated with these desirable attributes are imposed on the generalized algorithmic structure (*Algorithm* 1) is chosen as such to enable the representations of the free parameters ($\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5) in terms of the principal and spurious roots (ρ_∞ and ρ_∞^s) given by equation (3.3.83).

- In *Algorithm* 1, there is a total of 6 free parameters: $\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5. The analysis presented here, however, shows that there were 9 conditions needed to be fulfilled. An explanation for this difference is that the actual number of conditions being imposed directly on *Algorithm* 1 is only 6, i.e., 3 conditions for second order accuracy, 1 condition for zero-order overshoot behaviour, and 2 conditions for controllable numerical dissipation with the new selective control feature. By fulfilling these 6 conditions, the additional 3 conditions related to the constraints for achieving unconditional stability (equations (3.3.66) and (3.3.69)) are automatically satisfied.

We now demonstrate here that these constraints for achieving unconditional stability are indeed satisfied. Substituting equation (3.3.83) into the constraint for unconditional stability at low frequency limit (equation (3.3.66)) yields

$$\left|1 - \frac{1}{\Lambda_6 W_1}\right| = \left|1 - \frac{2(1+\rho_\infty)(1+\rho_\infty^s)}{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}\right| \tag{3.3.84}$$

Checking on the above value for a few representative algorithms:

- When $\rho_\infty = \rho_\infty^s = 1$, we have $\left|1 - \frac{2(1+\rho_\infty)(1+\rho_\infty^s)}{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}\right| = |-1| = 1$
- When $\rho_\infty = \rho_\infty^s = 0$, we have $\left|1 - \frac{2(1+\rho_\infty)(1+\rho_\infty^s)}{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}\right| = |\frac{1}{3}| = \frac{1}{3}$
- When $\rho_\infty = 1$ and $\rho_\infty^s = 0$, we have $\left|1 - \frac{2(1+\rho_\infty)(1+\rho_\infty^s)}{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}\right| = 0$

- When $\rho_\infty = 0.5$ and $\rho_\infty^s = 0.2$, we have $\left|1 - \frac{2(1+\rho_\infty)(1+\rho_\infty^s)}{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}\right| = |\frac{1}{5}| = \frac{1}{5}$

indicates that the constraint for achieving unconditional stability when the time step size (Δt) is taken to be very small i.e., at low frequency limit (equation (3.3.66)) is always satisfied. Likewise, substituting equation (3.3.83) into the two constraints for unconditional stability at high frequency limit (equation (3.3.69)) yields

$$\begin{aligned} \left|1 - \frac{\lambda_5}{\Lambda_5 W_2}\right| &= |-\rho_\infty| = \rho_\infty \\ \left|1 - \frac{\Lambda_4 W_1}{\Lambda_5 W_2}\right| &= |-\rho_\infty^s| = \rho_\infty^s \end{aligned} \tag{3.3.85}$$

Equation (3.3.85) indicates that the constraint for achieving unconditional stability in the limit of an infinite time step size i.e., at high frequency limit (equation (3.3.69)) is always satisfied since the maximum value for the two parameters is unity.

Having arrived at equation (3.3.83), we can now determine the corresponding weighted time field W used in the derivation process (equation (3.2.3)). Substituting equation (3.2.3) into equation (3.2.8), and choosing two expressions from equation (3.3.83), we can easily determine that the intermediate parameters w_1 and w_2 in equation (3.2.3) are given by

$$w_1 = -\frac{4(2+3\rho_\infty)}{1+3\rho_\infty}, \quad w_2 = \frac{10(1+\rho_\infty)}{1+3\rho_\infty} \tag{3.3.86}$$

Therefore, the corresponding weighted time field associated with the above algorithms design can be expressed in closed form as

$$W = 1 - \frac{4(2+3\rho_\infty)}{1+3\rho_\infty}\frac{t}{\Delta t} + \frac{10(1+\rho_\infty)}{1+3\rho_\infty}\left(\frac{t}{\Delta t}\right)^2 \tag{3.3.87}$$

Note that the above expression for the weighted time field W is not explicitly used in the computational implementation of the GS4-1 framework (it is given here only for

the reader's information). Such a procedure, presented in the next chapter, implicitly incorporates such a function through the algorithmic parameters $\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5 defined in equation (3.3.83). It is also worth mentioning that although W is independent of the spurious root ρ_∞^s as seen in equation (3.3.87), this parameter is contained in the approximations and design updates of the variables (for example, λ_5 is a function of ρ_∞^s, see equation (3.3.83)), hence its participation in the solution procedure of the developed framework.

The Algorithms by Design procedure is now completed, and we present in the next chapter the resulting GS4-1 framework with the desirable features including second-order time accuracy, unconditionally stability, zero-order overshoot, and controllable numerical dissipation with the new selective control feature, all of which are carefully imposed on the generalized algorithmic structure (*Algorithm* 1) via the Algorithms by Design procedure as described in this chapter. Relevant details including further description of the new selective control feature and a general guideline on how to use the new framework is also presented next.

Chapter 4

The New GS4-1 Time Integration Framework and the Relevant Details

This chapter presents the resulting GS4-1 framework, developed via the Algorithms by Design procedure in Chapter 3, and provides further relevant details to complete the description of the new framework and explain its implementation.

4.1 The New GS4-1 Time Integration Framework

Now that we have completed the Algorithms by Design procedure to design a family of algorithms within the GS4-1 framework with desirable features, we present the resulting framework in two forms: (i) the time derivative ($\dot{\boldsymbol{\phi}}$), and (ii) the primary variable ($\boldsymbol{\phi}$) forms of representation. That is, the unknowns to be solved are related to incremental quantities of $\dot{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$, respectively. In implementing and utilizing the framework for solving first order transient systems, one can employ either one of the forms of representation. Both forms yield the same results. The former has a more general and neat representation of the framework, while the latter is more convenient to work with for practical first order transient systems such as heat conduction problems.

4.1.1 The Time Derivative ($\dot{\phi}$) Form of Representation

Algorithm 2

The New GS4-1 Framework for Time Integration of First Order Transient System in $\dot{\phi}$-Form of Representation

Consider the linear first order transient system of governing equation and initial condition in the form

$$\mathbf{M}\dot{\boldsymbol{\phi}}(\mathrm{t}) + \mathbf{K}\boldsymbol{\phi}(\mathrm{t}) = \mathbf{F}(\mathrm{t}) \tag{4.1.1}$$

$$\boldsymbol{\phi}(0) = \boldsymbol{\phi}_0 \tag{4.1.2}$$

Employing the GS4-1 framework, the above governing equation can be represented in the incremental quantity of $\dot{\boldsymbol{\phi}}$ (i.e., $\dot{\boldsymbol{\phi}}$-form of representation) as follows

$$\mathbf{M}\widetilde{\dot{\boldsymbol{\phi}}} + \mathbf{K}\widetilde{\boldsymbol{\phi}} = \widetilde{\mathbf{F}} \tag{4.1.3}$$

where

$$\widetilde{\dot{\boldsymbol{\phi}}} = \dot{\boldsymbol{\phi}}_n + \Lambda_6 W_1 \Delta\dot{\boldsymbol{\phi}} \tag{4.1.4}$$

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_n + \Lambda_4 W_1 \Delta t\dot{\boldsymbol{\phi}}_n + \Lambda_5 W_2 \Delta t\Delta\dot{\boldsymbol{\phi}} \tag{4.1.5}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{4.1.6}$$

Therefore, given $\boldsymbol{\phi}_n$ and $\dot{\boldsymbol{\phi}}_n$, one can find $\boldsymbol{\phi}_{n+1}$ and $\dot{\boldsymbol{\phi}}_{n+1}$ by first solving for $\Delta\dot{\boldsymbol{\phi}}$ from

$$\begin{aligned}(\Lambda_6 W_1\mathbf{M} + \Lambda_5 W_2\Delta t\mathbf{K})\Delta\dot{\boldsymbol{\phi}} = &-\mathbf{M}\dot{\boldsymbol{\phi}}_n - \mathbf{K}(\boldsymbol{\phi}_n + \Lambda_4 W_1\Delta t\dot{\boldsymbol{\phi}}_n) \\ &+ \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n)\end{aligned} \tag{4.1.7}$$

followed by updating the variables at the end of each time level according to

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \lambda_4 \dot{\boldsymbol{\phi}}_n \Delta t + \lambda_5 \Delta \dot{\boldsymbol{\phi}} \Delta t \tag{4.1.8}$$

$$\dot{\boldsymbol{\phi}}_{n+1} = \dot{\boldsymbol{\phi}}_n + \Delta \dot{\boldsymbol{\phi}} \tag{4.1.9}$$

where

$$\begin{aligned} \Lambda_6 W_1 &= \frac{3 + \rho_\infty + \rho_\infty^s - \rho_\infty \rho_\infty^s}{2(1+\rho_\infty)(1+\rho_\infty^s)} \\ \Lambda_5 W_2 &= \frac{1}{(1+\rho_\infty)(1+\rho_\infty^s)} \\ \Lambda_4 W_1 &= \frac{1}{1+\rho_\infty} \\ W_1 &= \frac{1}{1+\rho_\infty} \\ \lambda_4 &= 1 \\ \lambda_5 &= \frac{1}{1+\rho_\infty^s} \end{aligned} \tag{4.1.10}$$

while ρ_∞ and ρ_∞^s are the principal and spurious roots at high frequency satisfying the following relation[1]

$$0 \leq \rho_\infty^s \leq \rho_\infty \leq 1 \tag{4.1.11}$$

4.1.2 The Primary Variable (ϕ) Form of Representation

Algorithm 3

The New GS4-1 Framework for Time Integration of First Order Transient System in ϕ-Form of Representation

Consider the linear first order transient system of governing equation and initial condition in the form

[1] see Section 3.3.4 for details

$$\mathbf{M}\dot{\boldsymbol{\phi}}(\mathrm{t}) + \mathbf{K}\boldsymbol{\phi}(\mathrm{t}) = \mathbf{F}(\mathrm{t}) \tag{4.1.12}$$

$$\boldsymbol{\phi}(0) = \boldsymbol{\phi}_0 \tag{4.1.13}$$

Employing the GS4-1 framework, the above governing equation can be represented in the incremental quantity of $\boldsymbol{\phi}$ (i.e., $\boldsymbol{\phi}$-form of representation) as follows

$$\mathbf{M}\widetilde{\dot{\boldsymbol{\phi}}} + \mathbf{K}\widetilde{\boldsymbol{\phi}} = \widetilde{\mathbf{F}} \tag{4.1.14}$$

where [2]

$$\widetilde{\dot{\boldsymbol{\phi}}} = \left(1 - \frac{\lambda_4\,\Lambda_6 W_1}{\lambda_5}\right)\dot{\boldsymbol{\phi}}_n + \frac{\Lambda_6 W_1}{\lambda_5\,\Delta t}\Delta\boldsymbol{\phi} \tag{4.1.15}$$

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta\boldsymbol{\phi} \tag{4.1.16}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{4.1.17}$$

Therefore, given $\boldsymbol{\phi}_n$ and $\dot{\boldsymbol{\phi}}_n$, one can find $\boldsymbol{\phi}_{n+1}$ and $\dot{\boldsymbol{\phi}}_{n+1}$ by first solving for $\Delta\boldsymbol{\phi}$ from

$$\begin{aligned}\left(\frac{\Lambda_6 W_1}{\lambda_5\Delta t}\mathbf{M} + \frac{\Lambda_5 W_2}{\lambda_5}\mathbf{K}\right)\Delta\boldsymbol{\phi} = &-\left(1 - \frac{\lambda_4\Lambda_6 W_1}{\lambda_5}\right)\mathbf{M}\dot{\boldsymbol{\phi}}_n - \mathbf{K}\boldsymbol{\phi}_n \\ &+ \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n)\end{aligned} \tag{4.1.18}$$

followed by updating the variables at the end of each time level according to

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \Delta\boldsymbol{\phi} \tag{4.1.19}$$

$$\dot{\boldsymbol{\phi}}_{n+1} = \left(1 - \frac{\lambda_4}{\lambda_5}\right)\dot{\boldsymbol{\phi}}_n + \frac{1}{\lambda_5\Delta t}\Delta\boldsymbol{\phi} \tag{4.1.20}$$

where the parameters $\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5 are defined in terms of the principal and spurious roots (ρ_∞ and ρ_∞^s) by equation (4.1.10).

[2] note that the following equations can easily be obtained after performing arithmetic operations on the equations in *Algorithm* 2

Remark 4.1.1

1. *Algorithms 2 and 3 are the resulting GS4-1 framework when the conditions for desirable algorithmic attributes (wish list) of second-order accuracy, unconditional stability, zero-order overshoot behaviour, and controllable numerical dissipation with the new selective control feature are imposed on the generalized structure of the algorithm design (Algorithm 1).*

2. *Algorithm 2 is the GS4-1 framework in $\dot{\boldsymbol{\phi}}$-form of representation while Algorithm 3 is the GS4-1 framework in $\boldsymbol{\phi}$-form of representation.*

3. *Algorithms 2 and 3 are controllable by two user-control parameters (ρ_∞ and ρ_∞^s) which represent the principal and spurious roots of the algorithm amplification matrix, respectively, at high frequency limit. These parameters are associated with the numerical damping of the primary variable and its time derivative, respectively, in this limit. Additionally, the selection for the values of these parameters must satisfy the following relation:* $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$

4. *By introducing the spurious root ρ_∞^s in addition to the principal root ρ_∞ in the design procedure, we are able to incorporate not only controllable numerical dissipative feature, but also with a more flexible and selective control of such a feature, while preserving second order accuracy and unconditional stability. This is in contrast to the classical Trapezoidal family of algorithms and the one-parameter controllable numerically dissipative method described in [62], which does not possess the selective control feature.*

5. *In Algorithms 2 and 3, the Crank-Nicolson method [60] is recovered for the selection of* $\rho_\infty = \rho_\infty^s = 1$.

6. *Algorithms 2 and 3 also recover the one-parameter controllable numerically dissipative method but without the selective control feature described in [62] when $\rho_\infty = \rho_\infty^s$.*

7. *Additionally, Algorithms 2 and 3 recover the Gear's method [63] when* $\rho_\infty = \rho_\infty^s = 0$ *[62].*

8. *With this new framework, the analyst has a precise and selective control of the high frequency damping associated with the primary variable and its time derivative through the two user-control parameters, ρ_∞ and ρ_∞^s, resulting in a*

whole family of time integration algorithms that can be tested for a particular problem at hand to yield physically accurate solutions of the problem. In this way, the two parameters, ρ_∞ and ρ_∞^s, may be increased or decreased according to the needs of the problem of interest (so long as the condition that $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$ is satisfied) until the resulting solutions are physically representative, i.e., (1) not too much oscillatory, and (2) stable without deviating from the true solutions, such that the dynamics of the problem is correctly represented. This process, therefore, needs to be done in a strategic manner and a general guideline on how to use the new framework to achieve the targeted objectives is given in Section 4.4.

9. *Since the designed framework is restricted to second order accuracy and does not consider first order conditions, the Forward Euler, Backward Euler, and Galerkin methods are purposely not to be recovered in Algorithms 2 and 3.*

4.1.3 Noteworthy Algorithms with $\rho_\infty = 1$

In this section we illustrate an interesting feature of the developed GS4-1 framework resulting from the selection of ρ_∞=1. Substituting this selection into *Algorithm* 3 while keeping ρ_∞^s variable yields

$$\left(\frac{\mathbf{M}}{\Delta t} + \frac{1}{2}\mathbf{K}\right)\boldsymbol{\phi}_{n+1} = \left(\frac{\mathbf{M}}{\Delta t} - \frac{1}{2}\mathbf{K}\right)\boldsymbol{\phi}_n + \frac{1}{2}(\mathbf{F}_{n+1} + \mathbf{F}_n) \tag{4.1.21}$$

$$\dot{\boldsymbol{\phi}}_{n+1} = (1+\rho_\infty^s)\left(\frac{\boldsymbol{\phi}_{n+1} - \boldsymbol{\phi}_n}{\Delta t}\right) - \rho_\infty^s \dot{\boldsymbol{\phi}}_n \tag{4.1.22}$$

An observation of equation (4.1.21) shows that the selection of ρ_∞=1 and any ρ_∞^s yields a family of algorithms that replicate the Crank-Nicolson method if only $\boldsymbol{\phi}$ is of interest. From equation (4.1.21), it is clear that for the selection of ρ_∞=1, $\dot{\boldsymbol{\phi}}$ does not participate in the solution procedure to determine $\boldsymbol{\phi}_{n+1}$. Therefore, although ρ_∞^s appears in the update equation of $\dot{\boldsymbol{\phi}}$, equation (4.1.22), it never affects the solution of $\boldsymbol{\phi}$; hence any selection of ρ_∞^s gives the same solution of $\boldsymbol{\phi}$. However, since the update of $\dot{\boldsymbol{\phi}}$ is a function of ρ_∞^s, different selection of ρ_∞^s results in different solution of $\dot{\boldsymbol{\phi}}$. In particular, the selection of ρ_∞^s=1 (and ρ_∞=1) exactly yields the Crank-Nicolson

method. On the other hand, the selections of any $\rho_\infty^s \neq 1$ yield a family of new algorithms with the selective control feature (all of which are unconditionally stable, second-order time accurate, and exhibits only zero-order overshoot) not existing in the past, that replicates the solutions of $\boldsymbol{\phi}$ of the Crank-Nicolson method, but result in different solutions of $\dot{\boldsymbol{\phi}}$ depending on the choice of the spurious root ρ_∞^s. This will be demonstrated using numerical examples in Sections 5.1 and 5.2. Of noteworthy importance is the particular algorithm belonging to this family, defined by selection of $\rho_\infty = 1$ and $\rho_\infty^s = 0$. The significant contributions of this new algorithm are demonstrated in some of the numerical examples presented in this thesis, whereby we show that this algorithm can already yield physically representative solutions of both variables even without having to impose numerical dissipation on the primary variable, i.e., with $\rho_\infty = 1$ in contrast to past/existing developments without the selective control feature whereby numerical dissipation is required to obtain physically representative solutions of these variables.

Algorithm 4
A Noteworthy Family of GS4-1 Algorithm, with Selection of $\rho_\infty = 1$***: Limiting Case [Crank-Nicolson Method (*** ρ_∞^s***=1)]***

Consider the linear first order transient system of governing equation and initial condition in the form

$$\mathbf{M}\dot{\boldsymbol{\phi}}(\mathrm{t}) + \mathbf{K}\boldsymbol{\phi}(\mathrm{t}) = \mathbf{F}(\mathrm{t}) \tag{4.1.23}$$

$$\boldsymbol{\phi}(0) = \boldsymbol{\phi}_0 \tag{4.1.24}$$

Given $\boldsymbol{\phi}_n$ *and* $\dot{\boldsymbol{\phi}}_n$*, one can find* $\boldsymbol{\phi}_{n+1}$ *from*

$$\left(\frac{\mathbf{M}}{\Delta t} + \frac{1}{2}\mathbf{K}\right)\boldsymbol{\phi}_{n+1} = \left(\frac{\mathbf{M}}{\Delta t} - \frac{1}{2}\mathbf{K}\right)\boldsymbol{\phi}_n + \frac{1}{2}(\mathbf{F}_{n+1} + \mathbf{F}_n) \tag{4.1.25}$$

and update $\dot{\boldsymbol{\phi}}_{n+1}$ *from*

$$\dot{\boldsymbol{\phi}}_{n+1} = (1+\rho_\infty^s)\left(\frac{\boldsymbol{\phi}_{n+1}-\boldsymbol{\phi}_n}{\Delta t}\right) - \rho_\infty^s \dot{\boldsymbol{\phi}}_n \tag{4.1.26}$$

Remark 4.1.2

1. *Algorithm 4 is the resulting family of algorithms in the GS4-1 framework in* $\boldsymbol{\phi}$*-form of representation (Algorithm 3) for particular selection of* $\rho_\infty = 1$*.*

2. *In Algorithm 4, the Crank-Nicolson method [60] is a special case recovered with the particular selection of* ρ_∞^s*=1.*

3. *In Algorithm 4, any selection of* ρ_∞^s *yields a family of algorithms that replicates the Crank-Nicolson method if only* $\boldsymbol{\phi}$ *is of interest; however it results in different solutions for* $\dot{\boldsymbol{\phi}}$ *leading to the design of a new family of GS4-1 algorithms with selective control feature, imposing no numerical dissipation on the primary variable (via* $\rho_\infty = 1$*), and all of which are second order accurate, unconditionally stable, and exhibit only a zero-order overshoot. Of noteworthy mention, is a particular new algorithm defined by choosing* $\rho_\infty = 1$ *and* $\rho_\infty^s = 0$*, which in some cases (as illustrated in Sections 6.2 and 6.3) can readily yield physically representative solutions of both the primary variable and its time derivative even without having to impose any numerical dissipation on the primary variable.*

4.2 Recovery of Existing Methods

Suppose we let $\rho_\infty^s = \rho_\infty$ in *Algorithms* 2 and 3. The algorithmic parameters (equation (4.1.10)) become

$$\begin{aligned} \Lambda_6 W_1 &= \frac{3-\rho_\infty}{2(1+\rho_\infty)}, \qquad \Lambda_5 W_2 = \frac{1}{(1+\rho_\infty)^2} \\ \Lambda_4 W_1 &= \frac{1}{1+\rho_\infty}, \qquad W_1 = \frac{1}{1+\rho_\infty} \\ \lambda_4 &= 1, \qquad \lambda_5 = \frac{1}{1+\rho_\infty} \end{aligned} \tag{4.2.27}$$

Substituting equation (4.2.27) into equations (4.1.4), (4.1.16) and (4.1.8) yields

$$\widetilde{\dot{\boldsymbol{\phi}}} = \dot{\boldsymbol{\phi}}_n + \left(\frac{3-\rho_\infty}{2(1+\rho_\infty)}\right)(\dot{\boldsymbol{\phi}}_{n+1} - \dot{\boldsymbol{\phi}}_n) \tag{4.2.28}$$

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_n + \left(\frac{1}{1+\rho_\infty}\right)(\boldsymbol{\phi}_{n+1} - \boldsymbol{\phi}_n) \tag{4.2.29}$$

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \dot{\boldsymbol{\phi}}_n \Delta t + \left(\frac{1}{1+\rho_\infty}\right)\Delta\dot{\boldsymbol{\phi}}\Delta t \tag{4.2.30}$$

Comparing equations (4.2.28) to (4.2.30) with the following corresponding equations from Reference [62],

$$\widetilde{\dot{\boldsymbol{\phi}}} = \dot{\boldsymbol{\phi}}_n + \alpha_m(\dot{\boldsymbol{\phi}}_{n+1} - \dot{\boldsymbol{\phi}}_n) \tag{4.2.31}$$

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_n + \alpha_f(\boldsymbol{\phi}_{n+1} - \boldsymbol{\phi}_n) \tag{4.2.32}$$

$$\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}_n + \dot{\boldsymbol{\phi}}_n \Delta t + \gamma\Delta\dot{\boldsymbol{\phi}}\Delta t \tag{4.2.33}$$

we can see that the three so-called free parameters in Reference [62] can be written here as

$$\alpha_m = \frac{3-\rho_\infty}{2(1+\rho_\infty)} \tag{4.2.34}$$

$$\alpha_f = \frac{1}{1+\rho_\infty} \tag{4.2.35}$$

$$\gamma = \frac{1}{1+\rho_\infty} \tag{4.2.36}$$

which agree with [62]. Therefore, the newly developed GS4-1 framework recovers an existing method with controllable numerical dissipation but without the new selective control feature described in [62] when $\rho_\infty^s = \rho_\infty$. This condition, as discussed previously in Section 3.3.4, indicates that the amount of the high frequency damping for the primary variable and its time derivative are imposed equally, in which case the selective control feature is turned off. This method is hence referred to, in this thesis, as the case without the selective control feature. Additionally, when $\rho_\infty^s = \rho_\infty = 1$,

the new framework recovers the nondissipative Crank-Nicolson method [60] while the Gear's method [63] is recovered if $\rho_\infty^s = \rho_\infty = 0$ [62].

4.3 The New Selective Control Feature

The key design step in the present development is the incorporation of a spurious root (ρ_∞^s), in addition to the principal root (ρ_∞), to allow for selective and a more flexible control of the high frequency damping for the primary variable and its time derivative (i.e., the selective control feature) in order to obtain physically representative solutions of those variables, thereby yielding a two-parameter (ρ_∞ and ρ_∞^s) family of algorithms. The selective control feature, which is new and is not available in any existing methods for first order transient system to-date, can be enabled by choosing $\rho_\infty \neq \rho_\infty^s$ so long as the condition that $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$ is satisfied. In this way, the analysts can select to impose certain amount of numerical damping for the primary variable, and impose another amount of numerical damping for the time derivative variable so as to obtain physically representative solutions of both variables that correctly capture the dynamics of the problem.

To demonstrate the effect of the selective control feature within the GS4-1 framework, we compare the results of the algorithms within the GS4-1 framework when the selective control feature is turned on and off. While having two parameters (ρ_∞ and ρ_∞^s) in the GS4-1 framework has a certain appeal, we recall that our aim is to simultaneously suppress the non-physical instabilities in both the primary variable and its time derivative to obtain solutions that are not only acceptable but also represent the dynamics of the problem correctly. For this purpose, we let ρ_∞^s in the case when the selective control feature is turned on to take a zero value regardless of the value of ρ_∞. Not only will this ensure the successful elimination of numerical oscillations associated with the time derivative variable which often requires much greater amount of numerical dissipation than the primary variable does (as demonstrated throughout this thesis), such an approach also allows for the widest range of ρ_∞ to be tested (due to the restriction that $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$). Given this constraint on ρ_∞^s value, the case when the selective control feature is turned on has only one parameter left to be specified (i.e., ρ_∞). For a valid comparison of the performance between the algorithms when the selective control feature is turned on and off, it is

necessary that the comparisons of results be made for the same value of ρ_∞ for the two cases. This way, the effect and significance of the new selective control feature can be demonstrated. The value of ρ_∞ may range from 1 (i.e., nondissipative/zero damping) to 0 (i.e., maximal damping).

Remark 4.3.1
To demonstrate the effect of the selective control feature within the GS4-1 framework, we compare the solutions of the following two cases of algorithms, given the same ρ_∞ values:

1. *the GS4-1 framework without the selective control feature (which recovers the current state-of-the-art method [62]), defined and denoted by: GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$). This algorithm is often referred to throughout the thesis as "the case without the selective control feature".*

2. *the GS4-1 framework with the selective control feature (which is new algorithm), defined and denoted by: GS4-1($\rho_\infty, \rho^s_\infty = 0$). Throughout the thesis, this algorithm is often referred to as "the case with the selective control feature".*

Together, the above are often referred to as "the two cases".

The results from all numerical examples considered in this study consistently indicate that the time derivative variable often requires more numerical damping than the primary variable does. Without the selective control feature, the solution of this variable is oscillatory and therefore does not represent the dynamics of the problem correctly. On the other hand, this requirement can easily be met via the selective control feature available in the present two-parameter GS4-1 framework, which provides a more flexible control of the high frequency damping of the two variables. The ability to generate physically accurate solutions of both the primary and time derivative variables via this new important feature serves as an added value and is a key desirable feature of the overall GS4-1 time integration framework; not to mention, second order accuracy, zero-order overshoot, unconditional stability, and with the least computational expense due to solving the system of equations only once and requiring the solutions of only one previous time step.

4.4 General Guidelines on How to Use the New Framework

In this section, we provide a general guideline on how to use the newly developed GS4-1 framework to solve a system of ordinary differential equations that is first order in time, after the matrices involved in the problem of interest (represented in general form as $\mathbf{M}$, $\mathbf{K}$, and $\mathbf{F}$) are formed using the chosen spatial discretization technique. According to *Algorithms* 2 and 3, to use the new GS4-1 framework to solve such a system of ODEs from the initial time t_0 to the end time t_{end} of the simulation, an analyst has to decide the values of: (1) the two parameters ρ_∞ and ρ_∞^s, and (2) the time step size Δt

4.4.1 Choosing the Values of the Two Parameters, ρ_∞ and ρ_∞^s

The fact that the algorithms within the new GS4-1 framework are defined by two parameters, ρ_∞ and ρ_∞^s means that the framework yields an infinite number of algorithms (so long as the condition $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$ is satisfied). An important question that arises is therefore what the optimal or appropriate values of these two parameters are for a particular problem of interest such that the resulting numerical solution best represents the dynamics of the problem. In this regard, an equally important fact to highlight is that in all research works involving the design and developments of time integration methods with controllable numerical dissipative feature, such as those described in [16, 26, 30, 62], the formula for such optimal values of the relevant algorithmic parameters are never provided. An explanation for this is that such optimal values are very problem-dependent; therefore, it would be very difficult if not impossible to formulate a generalized formula that works well for all problems. Hence, rather, the following guidelines on how to use the controllable numerically dissipative time integration framework is presented here.

In practice, one would typically solve any problem using nondissipative scheme ($\rho_\infty = \rho_\infty^s = 1$) first, since dissipative algorithms are only employed when the nondissipative counterpart fails to capture the problem physics. This is due to the fact that, as mentioned previously, numerical dissipation is an artifact added into the system to stabilize the algorithm in order to yield physically representative/accurate numerical solutions. Therefore, while dissipative algorithm is useful when the

nondissipative scheme fails, the use of it must be done in a strategic manner so as to avoid imposing over-dissipation. The importance of this delicate balancing act in imposing the numerical dissipation is demonstrated in this thesis, where we show in one of the numerical examples (see Section 6.2) how over-dissipation could yield numerical solutions that, although stable, do not represent the dynamics of the problem correctly.

The need for such a balancing act can easily be met by the new selective control feature inherent in the developed GS4-1 framework. We recall that our aim is to simultaneously suppress the non-physical instabilities in both the primary variable and its time derivative to obtain solutions that are not only acceptable but also represent the correct dynamics of the problem. Therefore, as previously mentioned in Section 4.3, we enforce ρ_∞^s in the GS4-1 framework to take on a zero value regardless of the value of ρ_∞ to ensure a successful elimination of the numerical oscillation associated with the time derivative variable as well as to allow for the widest range of ρ_∞ to be tested (due to the restriction that $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$). Given this constraint on ρ_∞^s value, there is only one parameter left to be specified (i.e., ρ_∞). Interestingly, in our investigations, we observed that in some cases, the GS4-1 framework can readily yield physically representative solutions of both the primary variable and its time derivative even without having to impose any numerical dissipation on the primary variable (i.e., $\rho_\infty = 1$) when $\rho_\infty^s = 0$, hence demonstrating the ability of the new selective control feature inherent in the GS4-1 framework. In this case, the GS4-1 framework yields a new algorithm defined by selection of $\rho_\infty = 1$ and $\rho_\infty^s = 0$ (see *Remark* 4.1.2). This noteworthy algorithm is to be tested if the totally nondissipative scheme ($\rho_\infty = \rho_\infty^s = 1$) fails to correctly capture the problem physics.

Typically, with the constraint of $\rho_\infty^s = 0$, the numerical solutions of the time derivative variable are found to be physically representative regardless of the value of ρ_∞. The quality of the numerical solutions of the primary variable, on the other hand, depends on the value of ρ_∞. In the case where $\rho_\infty = 1$ fails and results in highly oscillatory solution of the primary variable, dissipative algorithm ($\rho_\infty < 1$) is then used, keeping in mind the need for the balancing act as mentioned above. For this purpose, we first employ dissipative algorithm with only a slight amount of numerical dissipation (i.e.,

$\rho_\infty < 1$ but approaches unity). If the numerical solution of the primary variable is still not satisfactory, then more numerical dissipation is imposed until the physically representative solution is obtained, with a maximum possible numerical dissipation (only if necessary) imposed by choosing $\rho_\infty = 0$.

4.4.2 Choosing the Value of the Time Step Size Δt

In a numerical simulation involving dynamic systems, the accuracy of the resulting numerical solutions are dependent on the choice of the time step size Δt. A relatively small value of Δt ensures numerical solutions that better approach the exact solutions of the problem. However, this would require more computational time and memory. Hence, there is a tradeoff between accuracy and computational effort. In choosing the value of Δt that balances the two, a random value of Δt is initially chosen (typically the experience of the analyst in running simulations will become handy in choosing the initial value of Δt). After the numerical solutions are obtained, evaluation has to be made on whether or not the value of Δt should be further increased or decreased, so as to obtain physically representative solutions of the problem of interest.

To summarize, Figure 4.1 illustrates the general guidelines on how to solve a linear transient problem that is first order in time using the newly developed GS4-1 framework. For nonlinear problems, examples of the computational procedure involved in solving such systems are presented in Sections 6.2.4 and 6.3.5. Note that one could also use adaptive time-stepping.

4.5 Noteworthy Remarks on the Algorithm Designs and the Corresponding Time Level Issue

By introducing controllable numerical dissipation to the primary variable and its time derivative in the manner described above, the framework is subject to scrutiny on the time level evaluation of these variables. This issue on the time level of the two variables has implication on the construction of the proper convergence plot. Since the ability of the newly developed framework will be demonstrated partly through such a plot, which illustrates the order of accuracy of the algorithms, we find it important to bring up this time level issue for clarification purpose and for completeness of the discussion. It is worth mentioning that such an issue has been discussed and

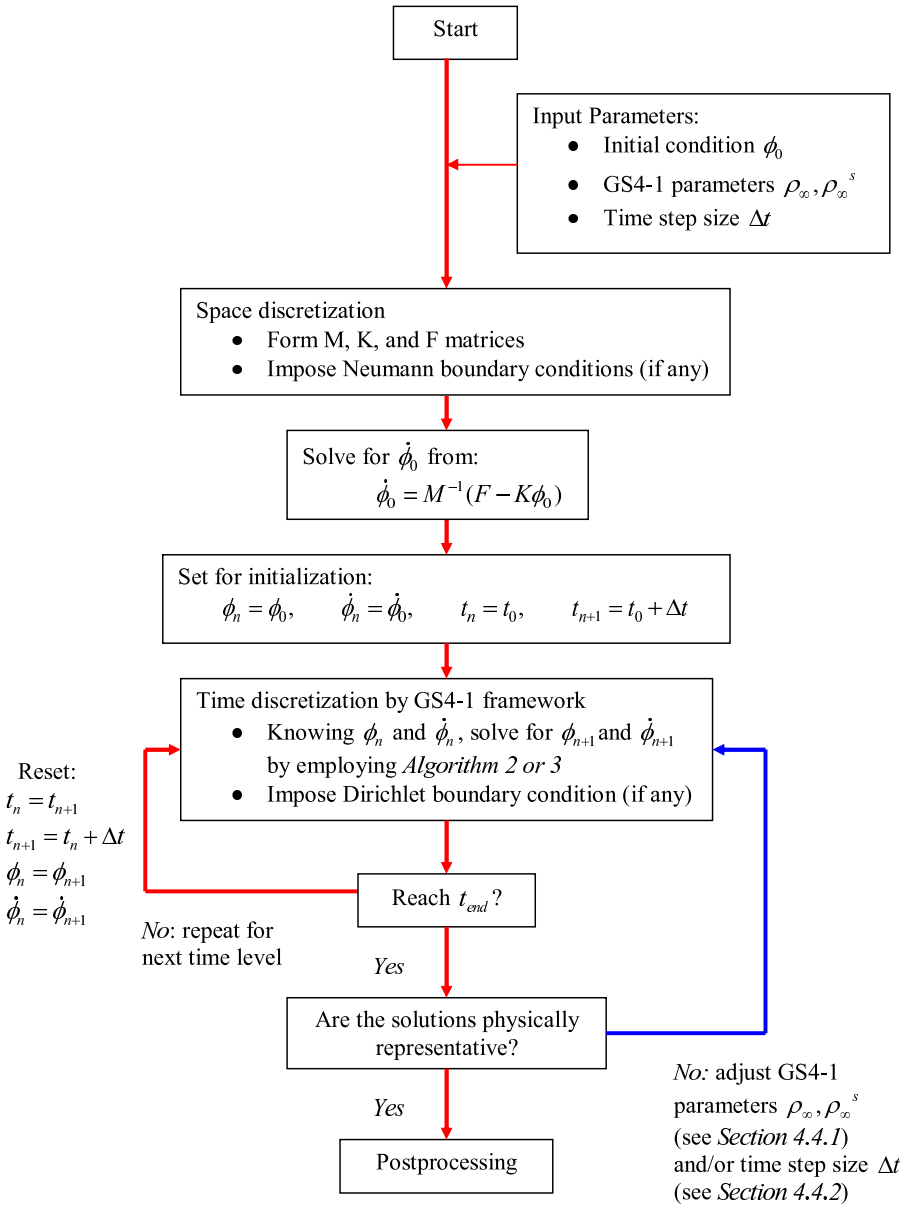


Figure 4.1: Illustration on how to solve a linear first order transient problem using the newly developed GS4-1 framework

illustrated for the structural dynamics cases (i.e., second order dynamic systems) in some earlier work [1, 73]. In this thesis, we focus on the discussion of the time level issues relevant to the algorithm designs for the first order transient systems, which is the scope of the present work.

4.5.1 The Correct Time Level for the Time Derivative Variable

We recall that the semi-discretized system of equations can be expressed in a temporally discrete form according to the GS4-1 framework as follows

$$\mathbf{M}\widetilde{\dot{\boldsymbol{\phi}}} + \mathbf{K}\widetilde{\boldsymbol{\phi}} = \widetilde{\mathbf{F}} \tag{4.5.37}$$

To see this time level issue more clearly, we express all variables in equation (4.5.37) in terms of their respective values between t_n and t_{n+1} time levels as follows:

$$\widetilde{\dot{\boldsymbol{\phi}}} = \dot{\boldsymbol{\phi}}_n + \Lambda_6 W_1(\dot{\boldsymbol{\phi}}_{n+1} - \dot{\boldsymbol{\phi}}_n) \tag{4.5.38}$$

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_n + \Lambda_4 W_1(\boldsymbol{\phi}_{n+1} - \boldsymbol{\phi}_n) \tag{4.5.39}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{4.5.40}$$

Equation (4.5.39) can be obtained by using the constraint for zero-order overshoot behaviour (that is, $\lambda_5\Lambda_4 W_1 = \Lambda_5 W_2$, see equation (3.3.59)) in equation (4.1.16).

Intuitively, all the quantities, namely, $\widetilde{\dot{\boldsymbol{\phi}}}$, $\widetilde{\boldsymbol{\phi}}$ and $\widetilde{\mathbf{F}}$ need to be calculated at the same time level for equation (4.5.37) to be physically meaningful. However, an observation of equation (4.1.10) indicates that $\Lambda_4 W_1 = W_1 \neq \Lambda_6 W_1$. It means that both $\widetilde{\boldsymbol{\phi}}$ and $\widetilde{\mathbf{F}}$ are interpolated linearly between time t_n and t_{n+1} at a specific time point $t_{n+\Lambda_4 W_1 \Delta t}$, whereas $\widetilde{\dot{\boldsymbol{\phi}}}$ appears to be interpolated at a different time level $t_{n+\Lambda_6 W_1 \Delta t}$ instead. To visualize this concept, the reader is directed to see Figure 4.2. The time-dependent load term ($\widetilde{\mathbf{F}}$) is interpolated in the same manner as the interpolation of $\widetilde{\boldsymbol{\phi}}$ (Figure 4.2(a)) since $W_1 = \Lambda_4 W_1$ (see equation (4.1.10)).

It is also worthy to note from equation (4.1.10) that $\Lambda_6 W_1 \Delta t \geq \Lambda_4 W_1 \Delta t$ for any Δt since

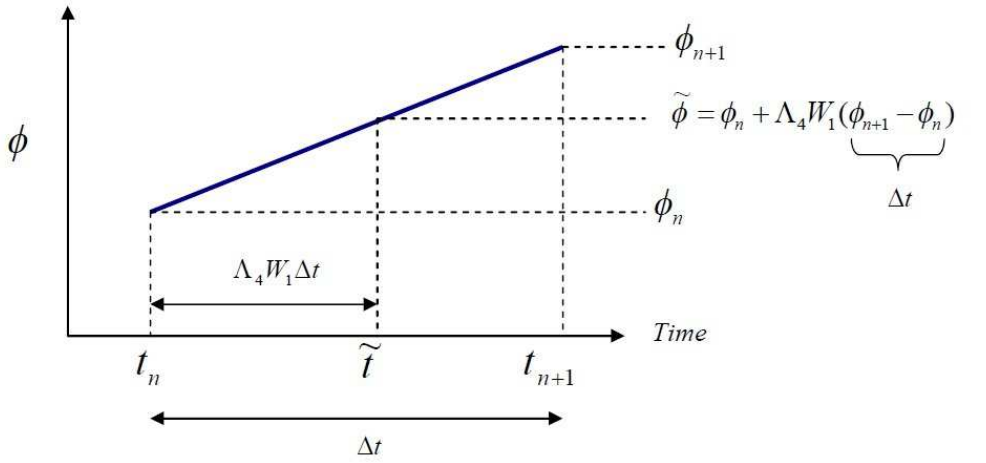
$$\begin{aligned}\Lambda_6 W_1 \Delta t - \Lambda_4 W_1 \Delta t &= \left(\frac{3 + \rho_\infty + \rho_\infty^s - \rho_\infty \rho_\infty^s}{2(1+\rho_\infty)(1+\rho_\infty^s)} - \frac{1}{1+\rho_\infty} \right) \Delta t \\ &= \frac{1-\rho_\infty^s}{2(1+\rho_\infty^s)} \Delta t \end{aligned} \tag{4.5.41}$$

is always either zero or a positive number for all $\rho_\infty^s \in [0, 1]$. Therefore, the time derivative variable ($\widetilde{\dot{\boldsymbol{\phi}}}$) may appear to be interpolated either: (i) at $\Lambda_4 W_1 \Delta t$ (i.e., at the same time level as $\widetilde{\boldsymbol{\phi}}$ and $\widetilde{\mathbf{F}}$), or (ii) at a time level forward from $\Lambda_4 W_1 \Delta t$ an amount of $\frac{1-\rho_\infty^s}{2(1+\rho_\infty^s)} \Delta t$ depending on the value of ρ_∞^s. The former is obtained when $\rho_\infty^s = 1$, whereas the latter is attained otherwise.

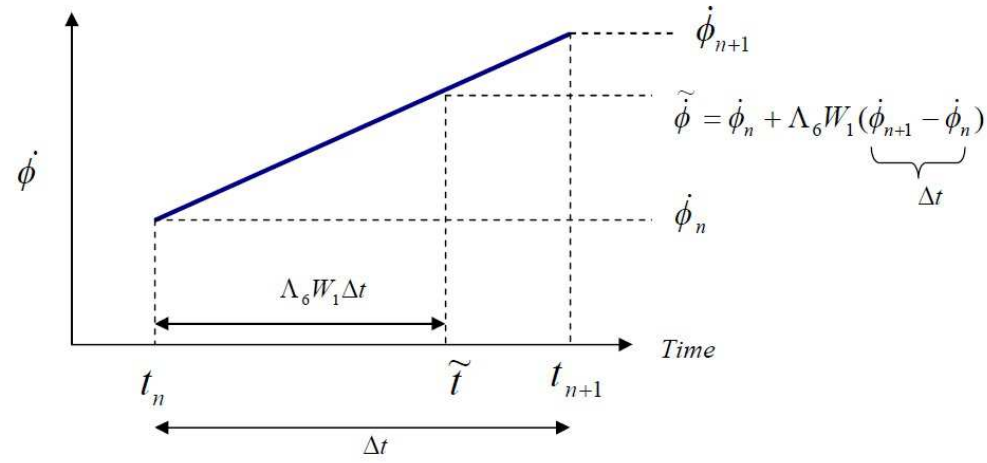
However, this seemingly inconsistent time level between these variables cannot be real since equation (4.5.37) requires that they are perfectly aligned at the same algorithmic time level for a physically meaningful equation. This conveys us one important information: a consistent time level between all these variables means that the t_{n+1} time level for the time derivative variable ($\dot{\phi}$) may not actually lie on this time level. It may be shifted back to a backward time level by an amount equal to $\frac{1-\rho_\infty^s}{2(1+\rho_\infty^s)} \Delta t$ as illustrated in Figure 4.3. Therefore, the correct t_{n+1} time level for $\dot{\phi}$ is given as

$$\begin{aligned} t_{n+1}^{\dot{\phi}} &= t_{n+1} - (\Lambda_6 W_1 - \Lambda_4 W_1) \Delta t \\ &= t_{n+1} - \frac{1-\rho_\infty^s}{2(1+\rho_\infty^s)} \Delta t \end{aligned} \tag{4.5.42}$$

An observation of equation (4.5.42) indicates that there will be no shift in $\dot{\phi}$ time level for ρ_∞^s value of unity. Under the same train of thought, it is apparent that the existing one-parameter method with the limited control of the high frequency damping is also subject to this shift in time level issue. We next discuss the effect of such a noteworthy issue.



(a)



(b)

Figure 4.2: A basic linear interpolation of the primary variable and its time derivative

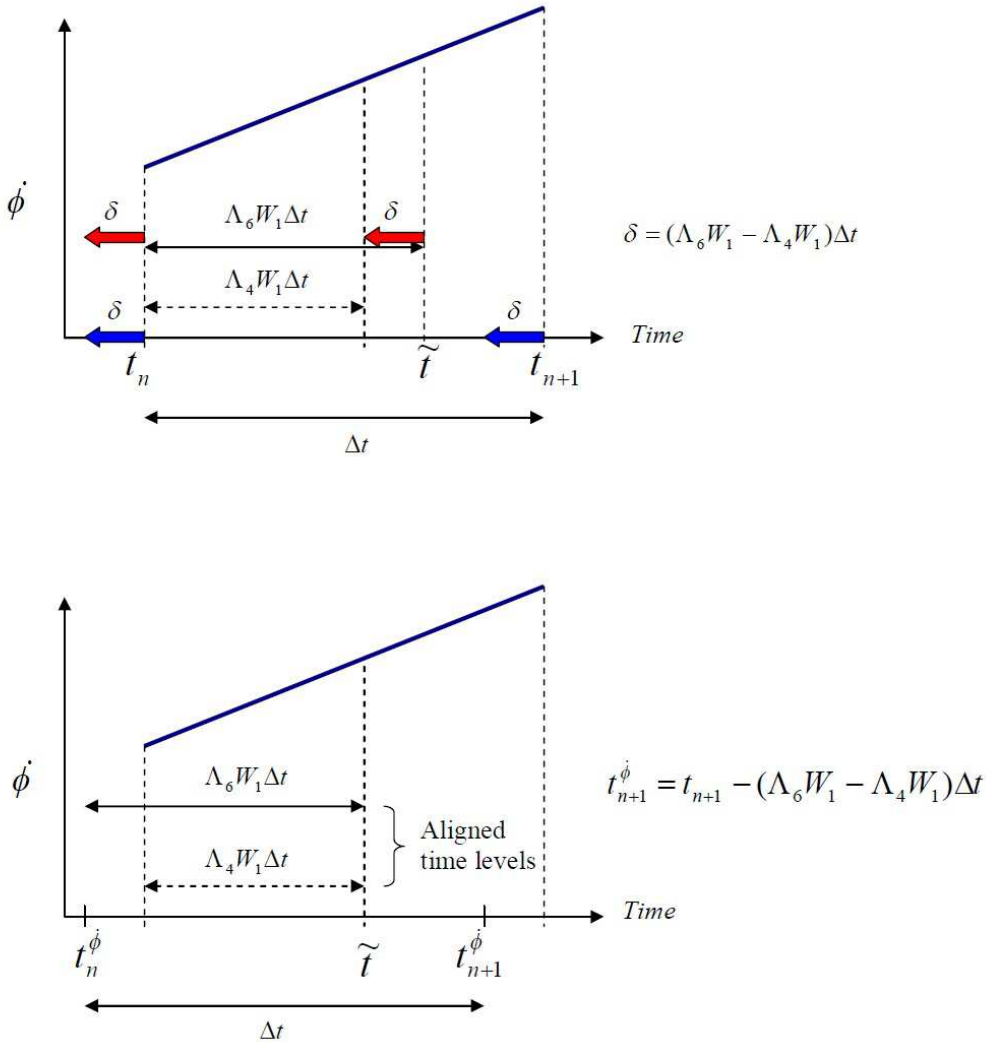


Figure 4.3: Time level alignment for the time derivative variable

4.5.2 The Effects on Proper Construction of the Convergence Plots

The shift in time level in the time derivative variable as described above has implication on the construction of the proper convergence plots as we shall discuss next (please refer to our previous expositions [1, 73] for a complete discussion on related matters for structural dynamics applications).

To better understand how this time level alignment issue affects the construction of the convergence plots, consider that we have four different time step sizes, $\Delta t_{i\,(i=1-4)}$, each of which is used to march the solution from t_0 to t_{end} for four separate runs. For the proper construction of the convergence plots, essentially we desire solutions at a specified time level, say t_{end} obtained using these time step sizes, to be compared to an 'exact' solution (usually taken as the reference solution obtained using a very small time step size at this same time level) for the computation of the errors. In other words, the computations of the errors require that the solutions are precisely evaluated at the same time level. The standard procedure for constructing the convergence plots randomly chooses the time step sizes without taking into account the possible shift in time level of the variables. It is worthy to note that because there is a possible shift in the time derivative variable ($\dot{\boldsymbol{\phi}}$) time level as discussed above and the shift is a function of the time step size Δt as seen in equation (4.5.42), the $\dot{\boldsymbol{\phi}}$ values obtained using the four different time step sizes may not actually be aligned at the same time level as illustrated in Figure 4.4(a). This may result in inaccurate computations of the errors and hence may yield a lower order of time accuracy (to be shown in Chapter 5). To circumvent this time level issue in constructing the proper convergence plots, we have previously defined and demonstrated an alternative way to contruct the plots such that the solutions of the variable with a time level shift are all aligned at the same time level, hence allowing for accurate computations of the errors. As illustrated in Figure 4.4(b), this so-called *time level aligned convergence plot* correctly aligns the time level for each case. Such a plot can be properly constructed by choosing the time step size for each run (Δt_i) such that the end time at which $\dot{\boldsymbol{\phi}}$ is evaluated is the same for all cases (i.e., $t_{end}^{\dot{\phi}}$ is correctly aligned). These time step sizes can be determined from equation (4.5.42) as follows. From equation (4.5.42), we have

$$t^{\dot{\phi}}_{end} = \Delta t_i(N_i - \Delta), \quad i = 1, ..., n \tag{4.5.43}$$

where $t^{\dot{\phi}}_{end}$ is the end time at which $\dot{\boldsymbol{\phi}}$ is evaluated (correctly aligned for all runs), Δt_i is the time step size for case i, N_i is the total number of time steps for case i, n is the total number of runs, and Δ is the shift in the $\dot{\boldsymbol{\phi}}$ time level given by

$$\Delta = \Lambda_6 W_1 - \Lambda_4 W_1 = \frac{1 - \rho^s_\infty}{2(1 + \rho^s_\infty)} \tag{4.5.44}$$

Therefore, the time level aligned convergence plot can be properly constructed for the time derivative variable by choosing the time step size for each case according to

$$\Delta t_i = \frac{t^{\dot{\phi}}_{end}}{N_i - \Delta}, \quad i = 1, ..., n \tag{4.5.45}$$

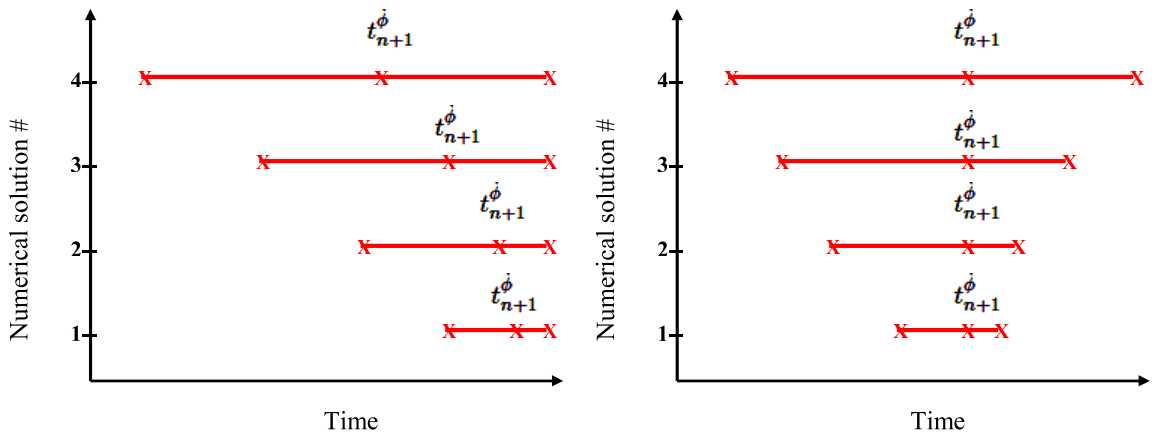


Figure 4.4: Visualization of final time level of $\dot{\boldsymbol{\phi}}$ utilizing: (a) standard convergence plot, and (b) time level aligned convergence plot [1]

Remark 4.5.1

1. *The standard convergence plot, in which the shift in $\dot{\boldsymbol{\phi}}$ time level is not taken into account, yields second order accuracy for the primary variable $\boldsymbol{\phi}$, whereas the time derivative variable $\dot{\boldsymbol{\phi}}$ is only first order accurate as a result of incorrect alignment of $\dot{\boldsymbol{\phi}}$ time level (due to the shift in $\dot{\boldsymbol{\phi}}$ time level).*

2. *Taking into account the shift in $\dot{\boldsymbol{\phi}}$ time level, the so-called time level aligned convergence plot properly yields second order accuracy for $\dot{\boldsymbol{\phi}}$ whereas $\boldsymbol{\phi}$ is only first order accurate (see Chapter 5). Intuitively, the correct alignment of $\dot{\boldsymbol{\phi}}$ time level also means that $\boldsymbol{\phi}$ time level is not correctly aligned.*

3. *More importantly, second order accuracy can be obtained for both $\boldsymbol{\phi}$ and $\dot{\boldsymbol{\phi}}$ if the time level for each of these variables are correctly aligned, that is, by properly constructing the standard convergence plot for $\boldsymbol{\phi}$ and the time level aligned convergence plot for $\dot{\boldsymbol{\phi}}$ (to take into account the shift in $\dot{\boldsymbol{\phi}}$ time level).*

4. *There is no shift in $\dot{\boldsymbol{\phi}}$ time level if $\rho_\infty^s = 1$, hence the two constructions of the convergence plot (namely the standard and the time level aligned convergence plots) yield the same results (second order accuracy for both $\boldsymbol{\phi}$ and $\dot{\boldsymbol{\phi}}$).*

4.6 Demonstration of the Algorithmic Attributes

Having discussed the desirable attributes of the newly developed GS4-1 framework imposed via the Algorithms by Design procedure in Section 3.3, for completeness we present in this section a demonstration of these algorithmic attributes. We first recall that the desirable attributes of the GS4-1 framework are second order accuracy, unconditional stability, zero-order overshoot, and controllable numerical dissipation with the new selective control feature. In this section, we only illustrate the unconditional stability and zero-order overshoot properties. The other two properties, namely second order accuracy and controllable numerical dissipation with selective control, will be discussed and illustrated throughout the thesis.

For the proof of unconditional stability, we consider again the SDOF model equation (equation (3.3.20)) and demonstrate the spectral radius plot of the algorithms within the GS4-1 framework for such a problem. To construct the spectral radius plot, we first determine the eigenvalues $\xi_{i\,(i=1,2)}$ of the algorithm amplification matrix $\mathbf{A}$. Recall from Section 3.3 that for this particular problem, $\mathbf{A}$ is given by equation (3.3.27) as follows

$$\mathbf{A} = \begin{bmatrix} 1 + \lambda_5 A_{21} & \lambda_4 - \lambda_5(1 - A_{22}) \\ \\ A_{21} & A_{22} \end{bmatrix} \tag{4.6.46}$$

where

$$A_{21} = -\frac{\Omega}{D} \tag{4.6.47}$$

$$A_{22} = 1 - \frac{1}{D}(1 + \Lambda_4 W_1 \Lambda) \tag{4.6.48}$$

$$D = \Lambda_6 W_1 + \Lambda_5 W_2 \Omega \tag{4.6.49}$$

$$\Omega = \lambda \Delta t \tag{4.6.50}$$

and the corresponding eigenvalues $\xi_{i\,(i=1,2)}$ are given by equation (3.3.62) as follows

$$\begin{aligned} \xi_{i\,(i=1,2)} =& \frac{1}{2}\left(2 - \frac{1}{D}[1 + (\lambda_5 + \Lambda_4 W_1)\Omega]\right) \\ & \pm \frac{1}{2}\sqrt{\frac{1}{D^2}[1 + 2(\lambda_5 + \Lambda_4 W_1)\Omega + (\lambda_5 + \Lambda_4 W_1)^2\Omega^2] - \frac{4}{D}\lambda_4 \Omega} \end{aligned} \tag{4.6.51}$$

Recall from Section 3.3.4 that the eigenvalues of the amplification matrix also represent the principal root ρ_∞ and the spurious root ρ_∞^s (the roots are denoted in general as ρ) of the algorithms[3]. The spectral radius plot can therefore be constructed by plotting these roots (ρ) as a function of $\lambda\Delta t$. By looking at the ranges of the roots at low and high $\lambda\Delta t$ limits as well as in between, the stability attribute of the algorithms can be assessed as discussed in Section 3.3.3.

For illustrative purpose, we consider $\lambda = 1$ and show the spectral radius plots of a number of algorithms within the new GS4-1 framework in comparison to those of the classical Trapezoidal family in Figure 4.5. For the classical Trapezoidal

[3]In particular, ρ_∞ represents the principal root of the algorithm amplification matrix at high frequency and is associated with the numerical dissipation of the primary variable in this limit. Meanwhile, ρ_∞^s represents the spurious root of the algorithm amplification matrix at high frequency and is related to the numerical dissipation of the time derivative variable in this limit

family, Figure 4.5(a) shows that only the Euler Backward ensures a smooth decay ($0 \leq \rho \leq 1$) which is non-oscillatory while both the Crank-Nicolson and Galerkin methods fall under the oscillatory decay range ($-1 \leq \rho < 0$) for large time step sizes (i.e., at high frequency limit). Although the resulting response may remain bounded, it is oscillatory and will not represent the true response realistically. The Euler Forward on the other hand, may yield results that grow with time and eventually becomes unbounded ($|\rho| > 1$). Although Euler Backward can give responses with smooth decay, it is worthy to recall that this method is only first order accurate. Figures 4.5(b)-(f) show the spectral radius plots of a number of algorithms within the GS4-1 framework with various ρ_∞ and ρ_∞^s values to represent the newly developed framework. An observation of these plots indicate that the GS4-1 algorithms are all unconditionally stable ($|\rho| \leq 1$). In addition, it is interesting to note that for combination of ρ_∞=1 and any ρ_∞^s, the resulting GS4-1 algorithm resembles the spectral radius plot of the Crank-Nicolson method if only the principal root is of interest as seen in Figures 4.5(b)-(d). As discussed in *Algorithm* 4, selecting $\rho_\infty = 1$ and any ρ_∞^s yields a family of algorithms that resemble the Crank-Nicolson method if only the primary variable is of interest. However, a different selection of ρ_∞^s results in different solutions of the time derivative variable with different time levels (recall from Section 4.5 that the actual time level of the time derivative variable is a function of ρ_∞^s, see equation (4.5.42)). In particular, the selection of ρ_∞^s=1 (and ρ_∞=1) yields exactly the Crank-Nicolson method where the time level of the time derivative variable is exactly at t_{n+1}. More interestingly, GS4-1 framework with the choice of $\rho_\infty = \rho_\infty^s = 0$ is L-stable (unconditionally stable with always a smooth decay, i.e., the roots always fall under the range of $0 \leq \rho \leq 1$) as seen in Figure 4.5(f). In contrast to the L-stable Euler Backward which is only first order accurate, the L-stable algorithm in the GS4-1 framework is second-order accurate. Note that we are able to not only introduce controllable numerical dissipation with the new selective control feature (in contrast to the existing one-parameter controllable numerically dissipative method without such a feature [62]), but we are also able to preserve the order of accuracy (in comparison to the classical Trapezoidal family). This highlights the clear improvement and advantage of the newly developed GS4-1 framework in comparison to the existing methods.

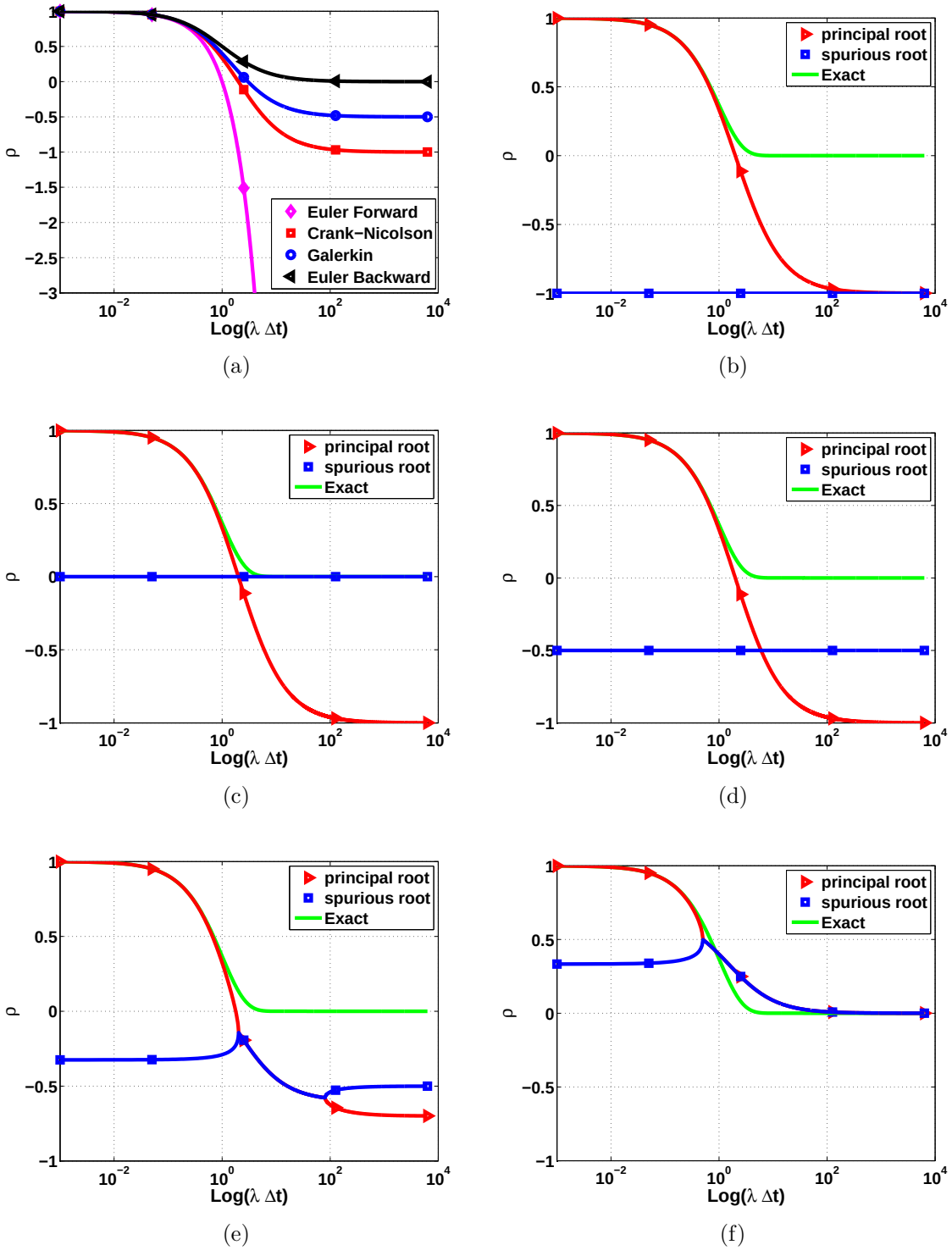


Figure 4.5: Spectral radius plots for the SDOF model problem with $\lambda = 1$: (a) Classical Trapezoidal family, (b) GS4-1($\rho_\infty, \rho^s_\infty$)=(1,1), (c) GS4-1($\rho_\infty, \rho^s_\infty$)=(1,0), (d) GS4-1($\rho_\infty, \rho^s_\infty$)=(1,0.5), (e) GS4-1($\rho_\infty, \rho^s_\infty$)=(0.7,0.5), (f) GS4-1($\rho_\infty, \rho^s_\infty$)=(0,0)

To demonstrate the zero-order overshoot behaviour on the primary variable, we again consider the SDOF model equation (equation (3.3.20)) with $\lambda = 1$ and $f = 0$. Since the order of overshoot is a function of the time step size Δt and can only be seen as the value of Δt increases at the high frequency limit, we consider here two different values of Δt representing high frequency: (1) $\Delta t = 100s$ and (2) $\Delta t = 1000s$ (note from Figure 4.5 that these time step sizes result in $\lambda \Delta t$ values that approach the high frequency limit for this model problem). We solve for the primary variable ϕ for initial value of $\phi_0 = 10$ and plot the solution as a function of the time level n. By comparing the solutions of ϕ generated using the two values of Δt, we can deduce on the algorithm order of overshoot on this variable. The illustrative plots of the overshoot behaviour of the representative algorithms within the GS4-1 framework (in particular, with $\rho_\infty^s = 0$) are shown in Figure 4.6 for randomly chosen ρ_∞ values: (a) $\rho_\infty = 0.9$, (b) $\rho_\infty = 0.7$, (c) $\rho_\infty = 0.4$, and (d) $\rho_\infty = 0.1$. From these plots, we can see that for all algorithms the amount of overshoot on the primary variable is relatively the same for both values of Δt. In other words, the amount of overshoot is independent of Δt. This demonstrates that the algorithms in the GS4-1 framework induce only zero-order overshoot behaviour on the primary variable, which is desirable especially in the context of large-scale practical nonlinear dynamic problems.

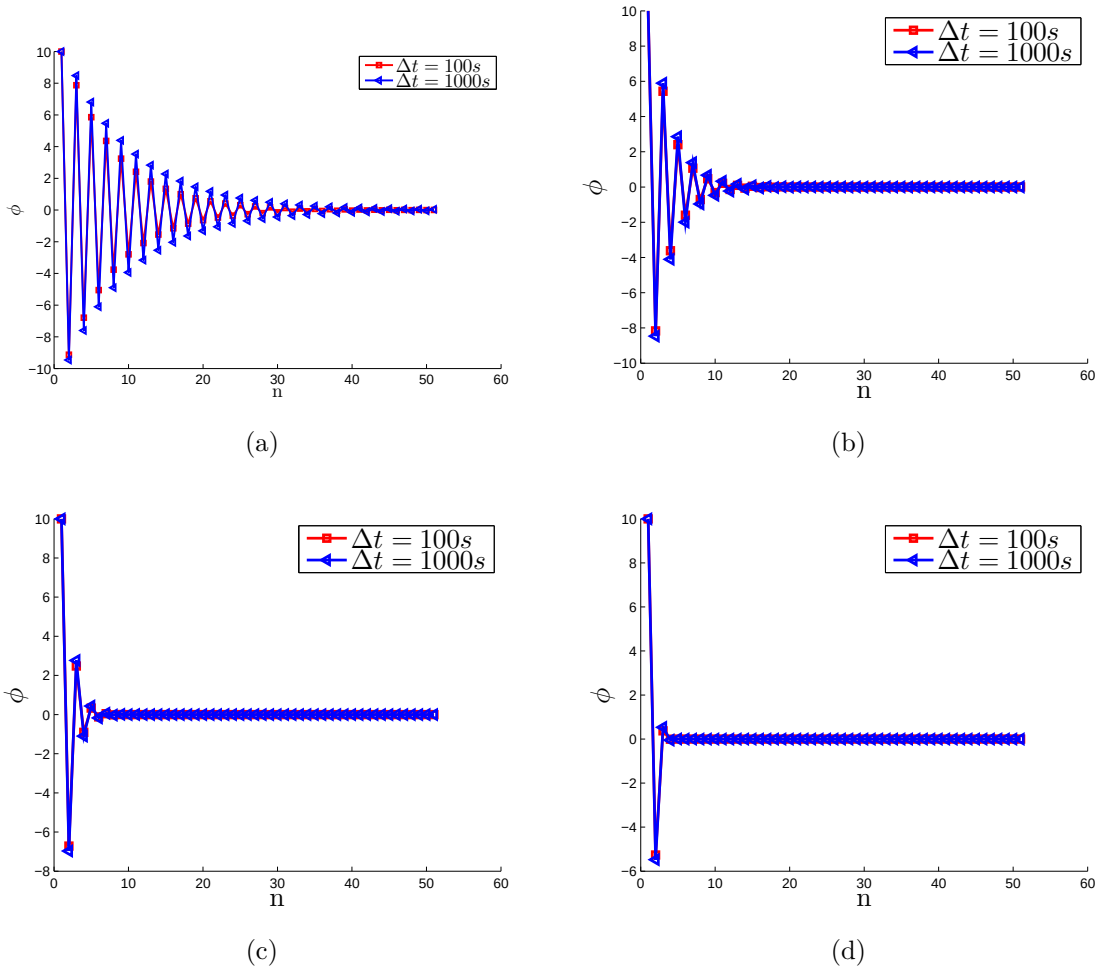


Figure 4.6: Illustrative overshoot behaviour of some randomly chosen algorithms within the GS4-1 framework (a) $(\rho_\infty = 0.9, \rho^s_\infty = 0)$, (b) $(\rho_\infty = 0.7, \rho^s_\infty = 0)$, (c) $(\rho_\infty = 0.4, \rho^s_\infty = 0)$, and (d) $(\rho_\infty = 0.1, \rho^s_\infty = 0)$

Chapter 5

Application of GS4-1 Framework to Linear First Order Transient Systems

As the main goal and contribution in this research is on the design and development of a new time integration framework, the most significant aspect of the research is upon investigating and demonstrating the ability and advantage of the new framework in comparison to the existing methods. Following this, it is necessary that the developed framework be tested on various types of numerical examples pertaining to first order transient systems to provide proof of consistency of the arguments on the ability of the new framework. Since the new GS4-1 framework was originally designed for applications in linear first order transient systems, we demonstrate in this chapter how this new framework with its computationally attractive features overcomes the restrictions in existing methods when applied to various linear first order systems. These include a simple Single Degree-of-Freedom (SDOF) example in Section 5.1, parabolic heat conduction in Section 5.2, flow transport in Section 5.3, and a convective heat transfer with thermal shock in Section 5.4. These numerical examples are purposely chosen as benchmark problems with available analytical solutions to allow for direct comparisons between the performance of the existing/current state-of-the-art methods and the newly developed GS4-1 framework. As a consequence to this, little attention is purposely given to the physics of the

benchmark problems with the intention to primarily highlight the performance of the time integration methods.

In demonstrating the computationally attractive features of the new framework through applications to various numerical examples, we recall that the stability and overshoot attributes of the newly developed GS4-1 framework have been demonstrated in the previous chapter. In this chapter and the following ones, we demonstrate the remaining two attributes of the framework, namely the accuracy and controllable numerical dissipation with the new selective control feature. Note again that the latter is the new and original contribution of the present development in comparison to the existing methods. For the demonstration of the ability of this new feature, on one hand we show that an equal amount of high frequency damping for the two variables (i.e, $\rho_\infty = \rho_\infty^s$) as in the existing and current state-of-the-art methods leads to non-physical instability in the time derivative variable for a minimal damping required to obtain acceptable solution of the primary variable. On the other hand, we demonstrate how this instability can easily be turned off via the new selective control feature (i.e, $\rho_\infty \neq \rho_\infty^s$) offered by our developed framework, thereby, demonstrating its ability and advantage over the existing methods.

5.1 Single Degree-of-Freedom (SDOF) Example

A Single Degree-of-Freedom (SDOF) numerical example is first presented to provide sole focus and discussions on time discretization without the need for spatial discretization arising for Multi-Degree-of-Freedom (MDOF) problems. Additionally, we highlight via this simple illustrative example the attributes of the newly developed GS4-1 framework in terms of the roles played by the two parameters, namely, the principal root ρ_∞ and the spurious root ρ_∞^s. These initial illustrations and discussions on the roles of these parameters are important to understand the consequent key attributes and contributions of the new framework in comparison to the current state-of-the-art method, which will be demonstrated and discussed in more detailed in the MDOF examples that follow throughout the thesis.

Consider a SDOF model equation in the form

$$\dot{y} + \lambda y = 10\cos(t/10) \tag{5.1.1}$$

$$y_0 = 100 \tag{5.1.2}$$

where the exact solution is given by

$$y(t) = \left(y_0 - \frac{1000\,\lambda}{1 + 100\,\lambda^2}\right) e^{-\lambda t} + \frac{100[10\,\lambda\, cos(t/10) + sin(t/10)]}{1 + 100\,\lambda^2} \tag{5.1.3}$$

We first demonstrate, using this illustrative example, our theoretical finding for $\rho_\infty = 1$ (see Section 4.1.3). Figure 5.1 shows the solution of the primary variable y as a function of time for $\lambda = 1$ solved using the GS4-1 family of algorithms with fixed $\rho_\infty = 1$ and a variety of ρ_∞^s values ranging from 0 to 1. First, it is worthy to recall that, when $\rho_\infty = \rho_\infty^s = 1$, the GS4-1 framework recovers the traditional Crank-Nicolson method. An observation of Figure 5.1 shows that the algorithms with the selection $\rho_\infty = 1$ and any ρ_∞^s values between 0 and 1 yield the same solution of y as that given by the Crank-Nicolson method (Figure 5.1(a)). This confirms our theoretical finding, as discussed before in Section 4.1.3, that for the selection of $\rho_\infty = 1$ and any ρ_∞^s, the developed GS4-1 framework yields a family of algorithms that replicates the Crank-Nicolson solutions if only the primary variable is of interest (see *Algorithm* 4 and *Remark* 4.1.2). Recall that this has also been proven by the spectral radius plot (Figure 4.5) whereby it can be seen that the spectral radius plots of the GS4-1 algorithms resulting from combination of ρ_∞=1 and any ρ_∞^s (Figure 4.5(b)-(d)) resembles that of the Crank-Nicolson method (Figure 4.5(a)).

Although the solutions of y are the same in this case (i.e, for the selection of $\rho_\infty = 1$), the solutions of $\dot{y}$ are different for different selection of ρ_∞^s as seen in Figure 5.2 which is to be expected. As discussed earlier in Section 4.1.3, since the update of $\dot{y}$ is a function of ρ_∞^s, different selection of ρ_∞^s results in a different solution of $\dot{y}$ and consequently a different time level of this variable. In particular, the selection of ρ_∞^s=1 (and ρ_∞=1) exactly yields the Crank-Nicolson method where the $\dot{y}$ time

level is exactly at t_{n+1}. This completely confirms our theoretical finding (see *Remark* 4.1.2) that, with the selection of ρ_∞=1 and any ρ_∞^s, the developed GS4-1 framework yields a family of different algorithms that replicates the solution of the primary variable generated by the Crank-Nicolson method but yields different solution of the time derivative variable (with its time level ranging between $t_{n+1/2}$ and t_{n+1}), all of which are unconditionally stable, second-order accurate, and exhibit only a zero-order overshoot behaviour.

We next illustrate, using this example, the roles played by the two parameters (i.e., the principal root ρ_∞ and the spurious root ρ_∞^s) as this is important to understand the consequent key attributes and contributions of the new GS4-1 framework in comparison to the current state-of-the-art method (which will be demonstrated and discussed in more detailed in the MDOF examples that follow throughout the thesis). For this purpose, we first show the effect of ρ_∞ by solving the problem using the GS4-1 framework with arbitrarily chosen value of $\rho_\infty^s = 0.1$ and a variety of ρ_∞ values ranging from 0.9 to 0.2 (the lower boundary was carefully chosen to satisfy the relation that $\rho_\infty^s \leq \rho_\infty$, see *Remark* 3.3.1). The solutions of the primary variable and its time derivative (y and $\dot{y}$) are shown in Figures 5.3 and 5.4, respectively. As seen in Figure 5.3, there is a clear difference in the solution of y generated by the considered algorithms and this is due to the influence of ρ_∞ value in the computation of y. Meanwhile, Figure 5.4 shows only a slight difference[1] in the solution of $\dot{y}$. The above analysis demonstrates that ρ_∞ has dominant influence on the numerical solution of the primary variable, but cause only slight effect on the numerical solution of the time derivative variable.

Remark 5.1.1

1. *In general, for the same value of ρ_∞^s and different value of ρ_∞, there is a clear difference in the solution of the primary variable (see Figure 5.3).*

2. *On the other hand, there is only a slight difference in the solution of the time derivative variable for the same value of ρ_∞^s and different value of ρ_∞ (see Figure 5.4).*

[1] Although ρ_∞^s, which is associated with the numerical dissipation property of $\dot{y}$, is kept constant in this case, there is a subtle difference in the solution of $\dot{y}$ because the computation of this variable is influenced by the solution of y which is dependent on ρ_∞ value.

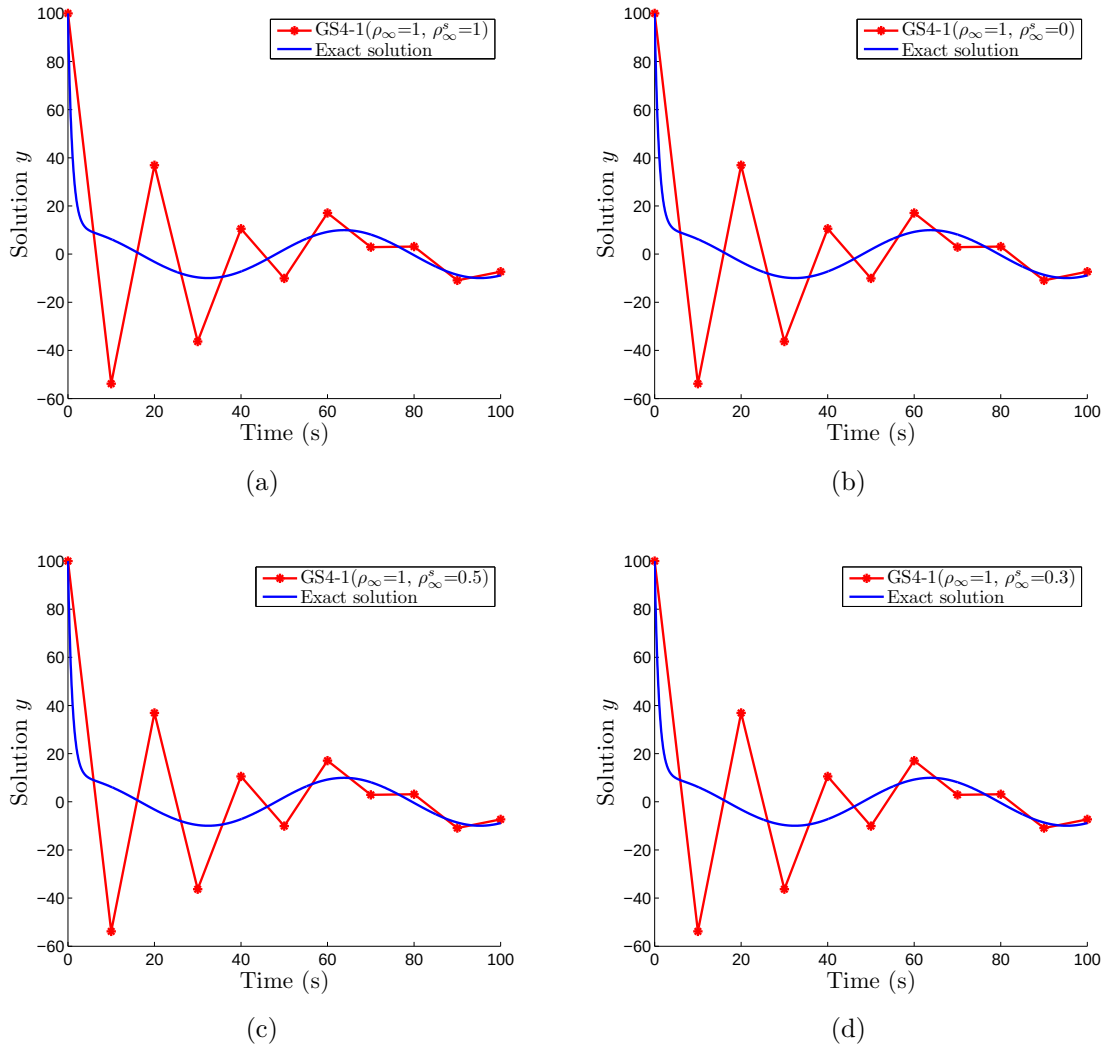


Figure 5.1: y (for the SDOF Example, section 5.1) as a function of time solved using: (a) Crank-Nicolson method / GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0.5)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0.3)$, all of which solved using a time-step size $\Delta t = 10$s and are compared to the exact solution given by equation (5.1.3)

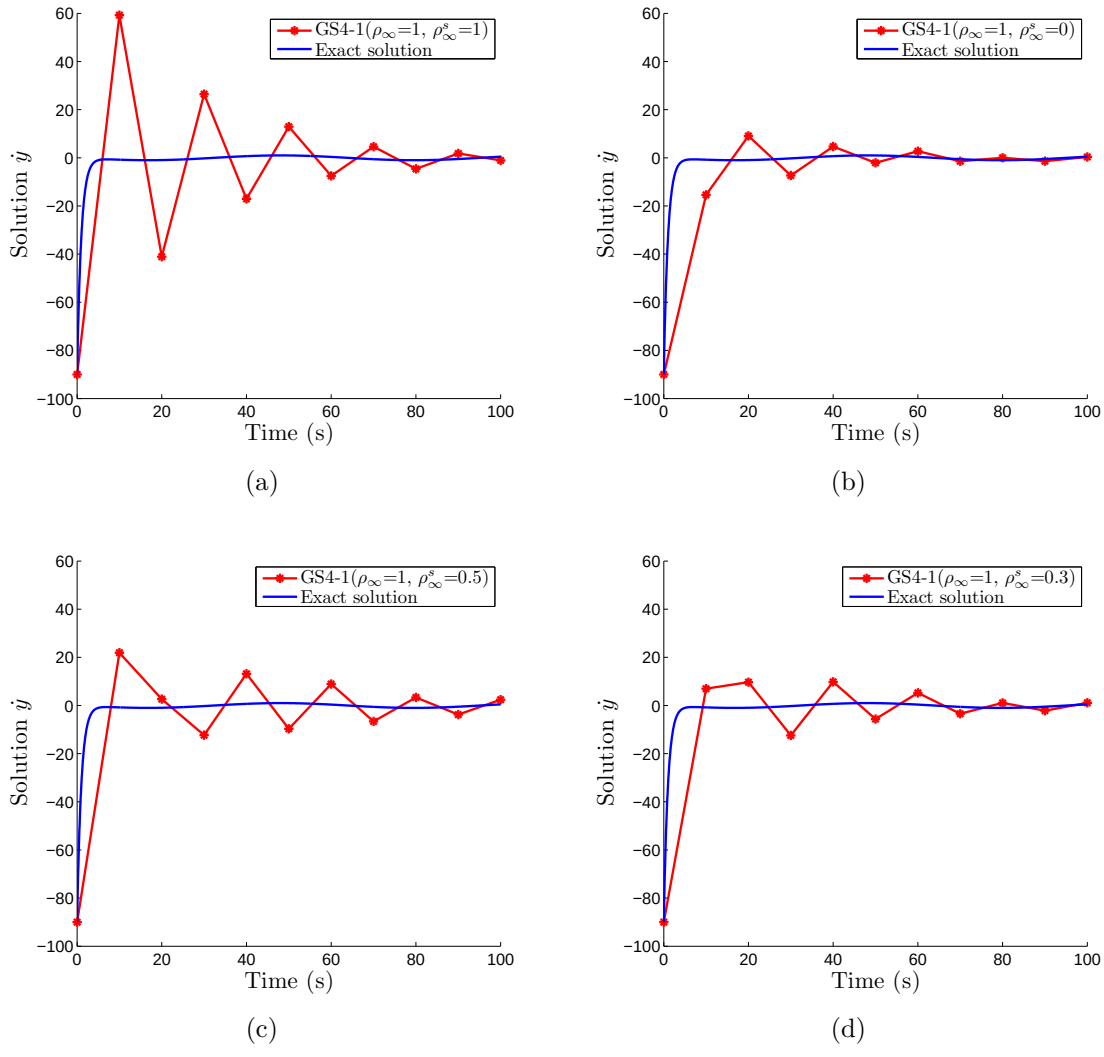


Figure 5.2: $\dot{y}$ (for the SDOF Example, section 5.1) as a function of time solved using: (a) Crank-Nicolson method / GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0.5)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0.3)$, all of which solved using a time-step size $\Delta t = 10$s and are compared to the exact solution given by equation (5.1.3)

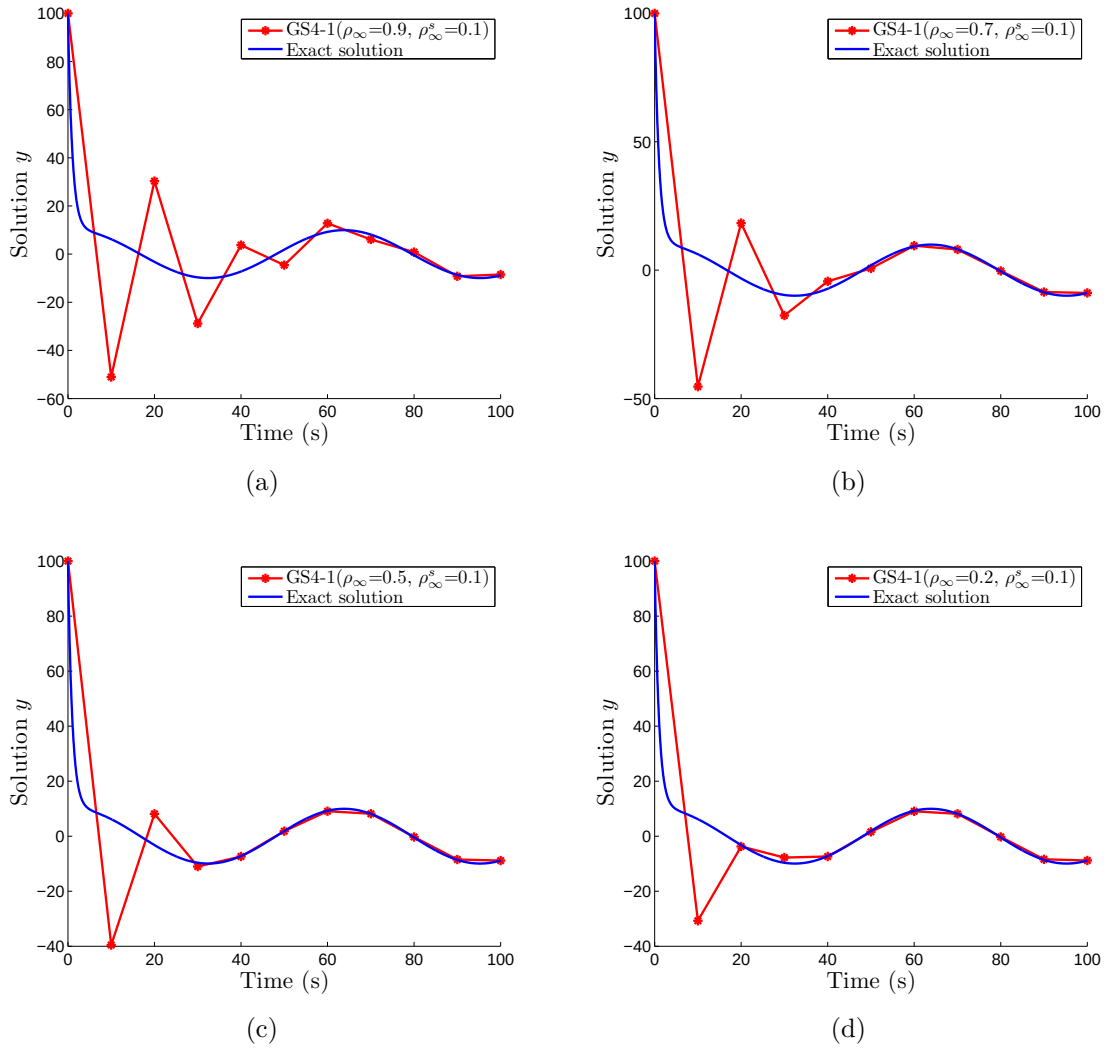


Figure 5.3: y (for the SDOF Example, section 5.1) as a function of time solved using: (a) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.9,0.1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.1)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.5,0.1)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.2,0.1)$, all of which solved using a time-step size $\Delta t = 10$s and are compared to the exact solution given by equation (5.1.3)

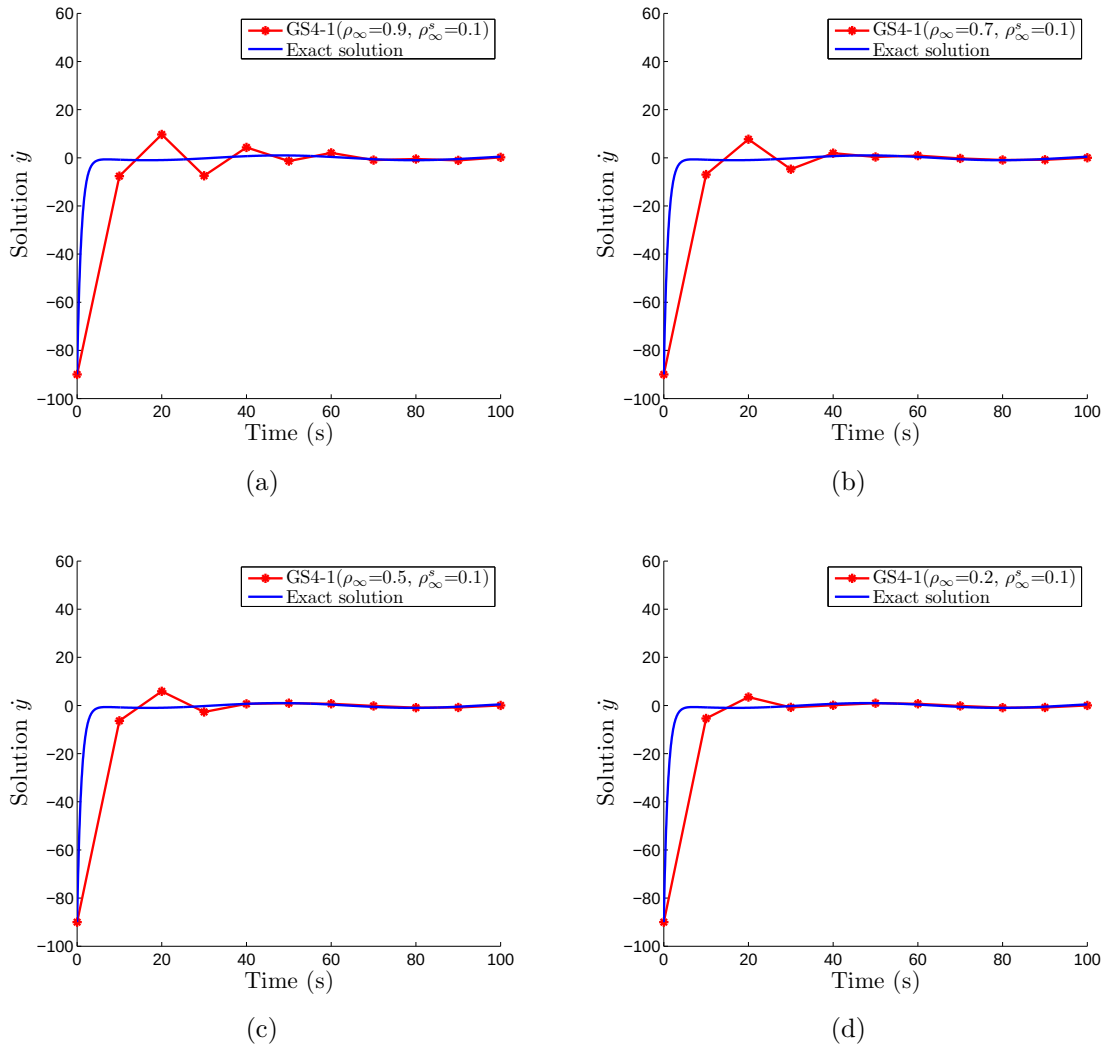


Figure 5.4: $\dot{y}$ (for the SDOF Example, section 5.1) as a function of time solved using: (a) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.9,0.1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.1)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.5,0.1)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.2,0.1)$, all of which solved using a time-step size $\Delta t = 10$s and are compared to the exact solution given by equation (5.1.3)

3. *That is, the principal root ρ_∞ has dominant influence on the numerical solution of the primary variable, but has only slight effect on the numerical solution of the time derivative variable.*

4. *From the above observations, it is to be noted that the two parameters are not completely controlling the numerical dissipation of the two variables separately. In other words, the decoupling of one another is not complete.*

We show next the effect of the spurious root ρ_∞^s by solving the problem using algorithms within the GS4-1 framework with an arbitrarily chosen value of $\rho_\infty = 0.7$ and a variety of ρ_∞^s ranging from 0.7 to 0 (the upper boundary was carefully chosen to satisfy the relation that $\rho_\infty^s \leq \rho_\infty$). The solutions of the two variables are shown in Figures 5.5 and 5.6, respectively. As seen in Figure 5.5, there is only a subtle difference in the solution of y owing to the slight influence of ρ_∞^s value in the computation of y (note that this is valid only for the case of $\rho_\infty \neq 1$. Recall that for the case of $\rho_\infty = 1$, any ρ_∞^s value yields the same solution of y as discussed and demonstrated before). However, Figure 5.6 shows a clear difference in the solutions of $\dot{y}$ for the different values of ρ_∞^s as expected. This demonstrates that ρ_∞^s greatly influences the numerical solution of the time derivative variable and has only subtle effect on the numerical solution of the primary variable.[2]

Remark 5.1.2

1. *In general, for the same value of ρ_∞, different value of ρ_∞^s results in only subtle differences in the solution of the primary variable (see Figure 5.5).*

2. *For different values of ρ_∞^s, however, there is a clear difference in the solution of the time derivative variable for the same value of ρ_∞ (see Figure 5.6).*

3. *That is, ρ_∞^s has dominant effect on the numerical solution of the time derivative variable, but has only subtle influence on the numerical solution of the primary variable.*

Additionally, we can see from Figures 5.3 and 5.6 that in general the oscillation in the numerical solution of each variable is reduced as the value of the dominant

[2]However, when $\rho_\infty = 1$, ρ_∞^s has no effect at all on the numerical solution of the primary variable as discussed before (see Section 4.1.3 and also Figure 5.1)

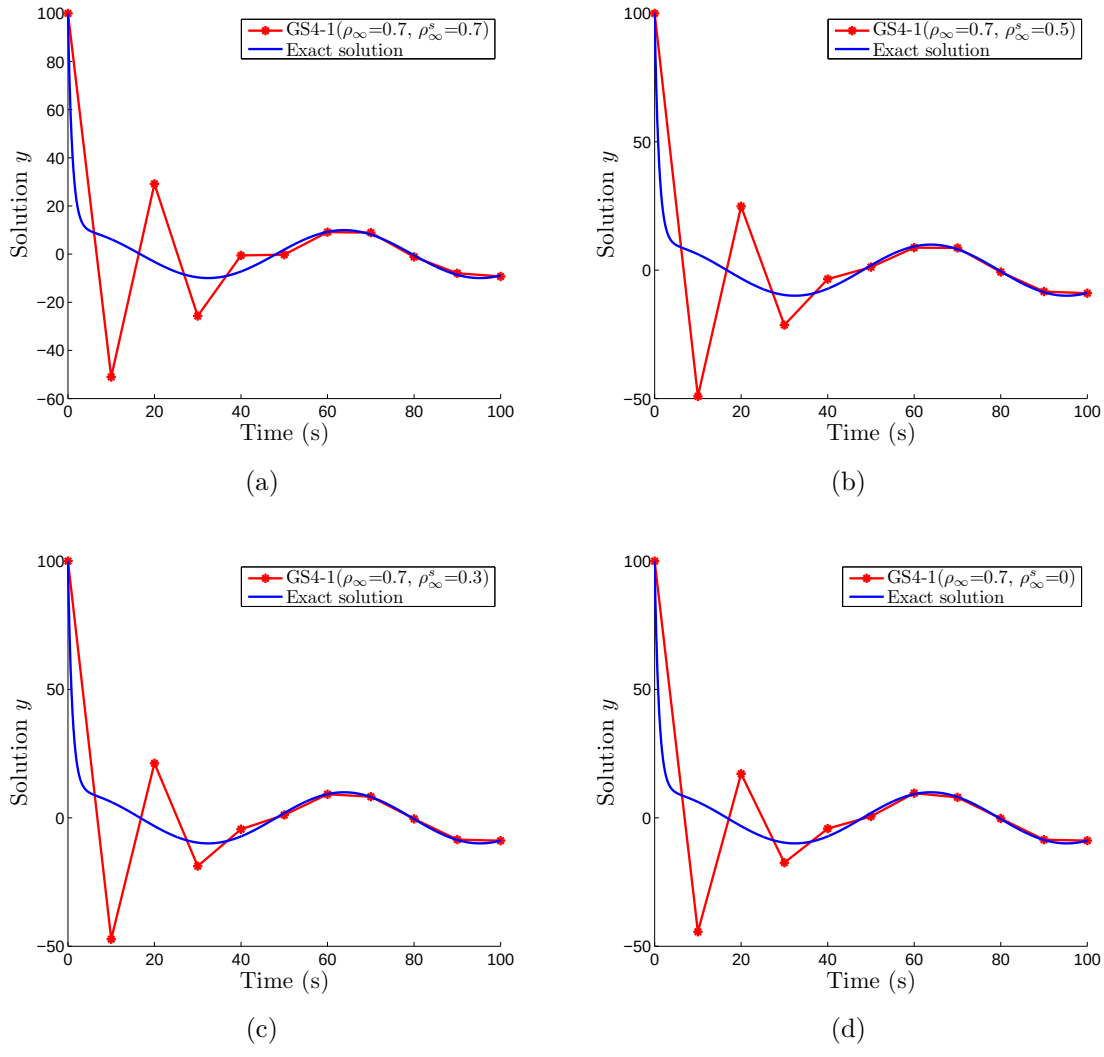


Figure 5.5: y (for the SDOF Example, section 5.1) as a function of time solved using: (a) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.7)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.5)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.3)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.0)$, all of which solved using a time-step size $\Delta t = 10$s and are compared to the exact solution given by equation (5.1.3)

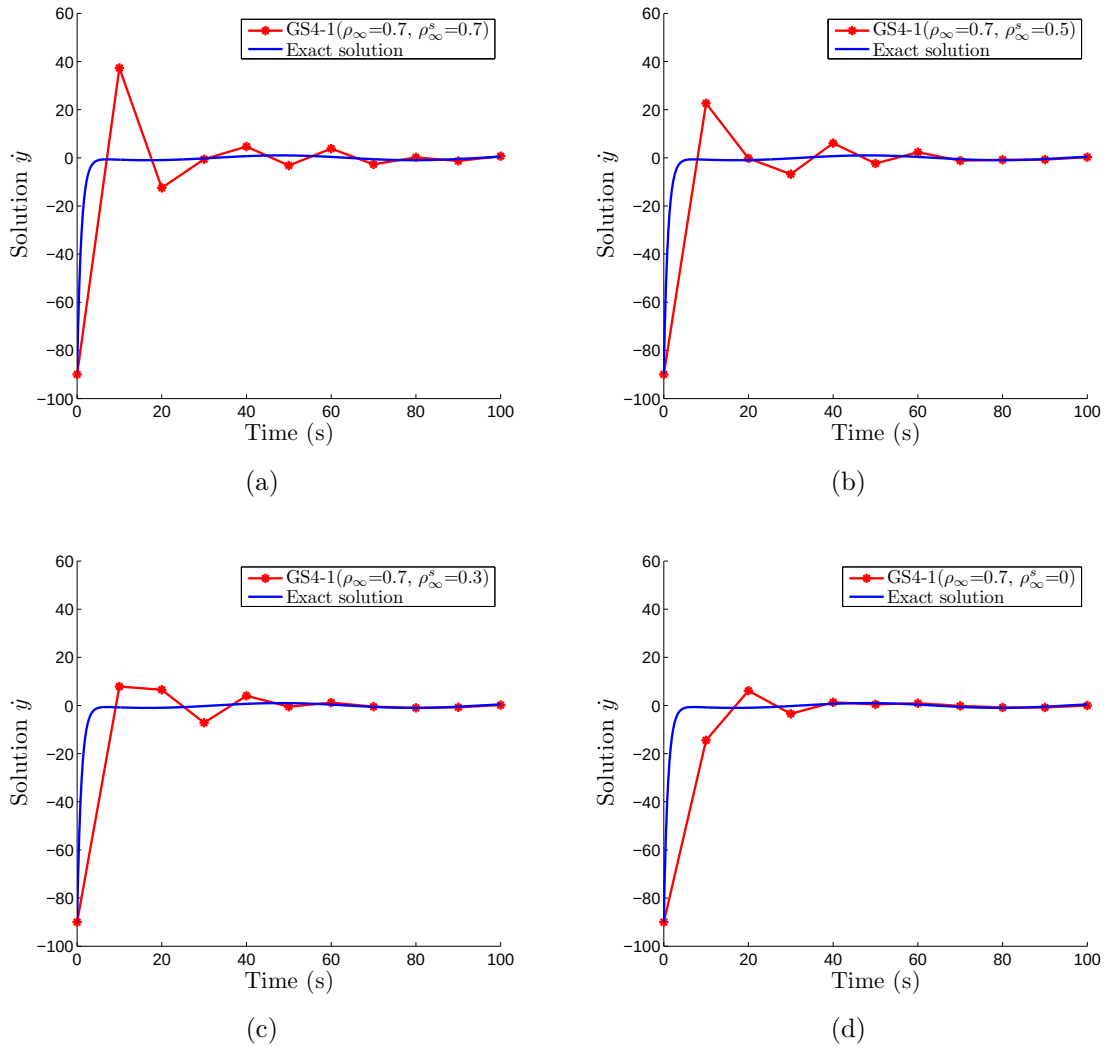


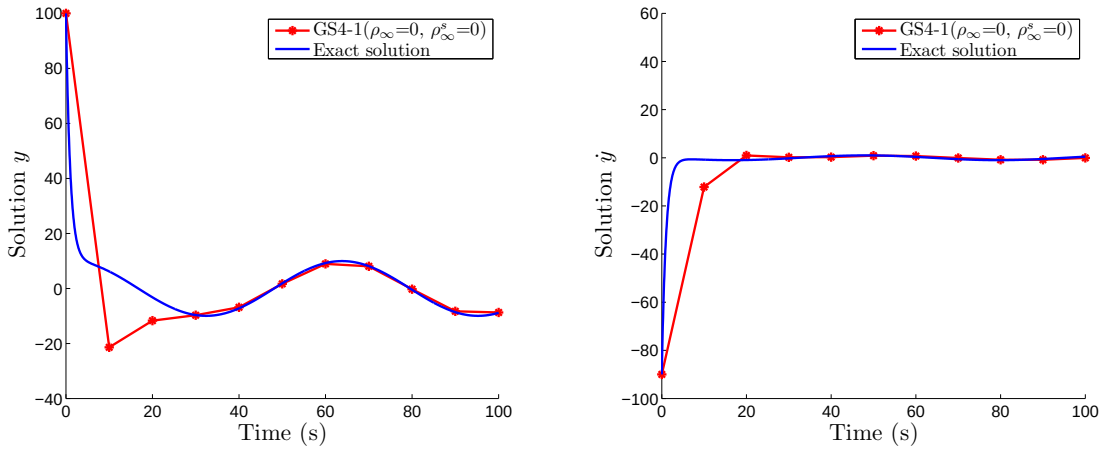
Figure 5.6: $\dot{y}$ (for the SDOF Example, section 5.1) as a function of time solved using: (a) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.7)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.5)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.3)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.0)$, all of which solved using a time-step size $\Delta t = 10$s and are compared to the exact solution given by equation (5.1.3)

parameter for the respective variable is reduced. The decrease in the oscillation is the effect of imposing more numerical dissipation on the variable via the selection of smaller value of the dominant parameter (recall that a parameter value of unity means no numerical dissipation (i.e., nondissipative/zero damping) is imposed on the respective variable, while a value of zero imposes maximum numerical dissipation on the variable, see *Remark* 3.3.1). This demonstrates that numerical dissipation is an important property of an algorithm that serves to stabilize the algorithm in order to yield physically representative numerical solutions.

From the analysis presented above, we can see the roles played by the two parameters ρ_∞ and ρ_∞^s in the developed GS4-1 framework, which demonstrate one important information; that the two parameters selectively control the numerical dissipation of the primary variable and its time derivative, respectively, i.e., the selective control feature which is the new attribute of the developed GS4-1 framework in comparison to existing methods to-date. This feature can be enabled by selecting $\rho_\infty \neq \rho_\infty^s$ so long as the condition that $0 \leq \rho_\infty^s \leq \rho_\infty \leq 1$ is satisfied. On the other hand, by selecting $\rho_\infty = \rho_\infty^s$, the selective control feature is turned off and the framework naturally recovers the current state-of-the-art method (without such a feature) [62]. This new feature enables the analysts to impose different amounts of numerical dissipation on the two variables by choosing $\rho_\infty \neq \rho_\infty^s$ to obtain successful simultaneous elimination of the numerical oscillations associated with these two variables towards obtaining physically representative solutions of the problem. This is in contrast to the existing method without such a feature (i.e., $\rho_\infty = \rho_\infty^s$), which often yields numerical instabilities in the time derivative variable. This is not desirable since the time derivative variable may have practical importance in some experimental work [64, 65]; therefore, obtaining solutions that are representative is important in this regard. This requirement can easily be met by the GS4-1 framework via the new selective control feature. Such an ability of the newly developed framework will be demonstrated in the following numerical examples throughout the thesis.

It is also worthy to recall that the developed GS4-1 framework can recover the L-stable feature contained in the existing Euler Backward and Gear's methods by choosing $\rho_\infty = \rho_\infty^s = 0$ (see Figure 4.5). Figure 5.7 shows the solutions (y and

$\dot{y}$) obtained using GS4-1 with (ρ_∞,ρ^s_∞) = (0,0) which is the L-stable algorithm contained in the GS4-1 framework. It is also worth mentioning that although the classical Euler Backward method is L-stable, it is only first order accurate while the present development is second order accurate and hence the obvious improvement in comparison to the Euler Backward method.



(a) (b)

Figure 5.7: The solution of SDOF Example (section 5.1) as a function of time solved using GS4-1 with (ρ_∞,ρ^s_∞) = (0,0) with a time-step size Δt = 10s and are compared to the exact solution given by equation (5.1.3): (a) y, (b) $\dot{y}$

Remark 5.1.3

1. *With particular selection of $\rho_\infty = \rho^s_\infty = 0$, the developed GS4-1 framework naturally recovers an L-stable algorithm (see also Figure 4.5).*

2. *The L-stable algorithm is second order accurate, and hence has a clear advantage over the classical Euler Backward method which is only first order accurate (although it is also L-stable).*

Just to give a flavour of the resulting solutions when the time step size Δt is taken to be small enough, we solve the problem using few algorithms within the GS4-1 framework: (a) $\rho_\infty = \rho^s_\infty = 1$ (i.e., the traditional Crank-Nicolson method), (b) $\rho_\infty = 1$,$\rho^s_\infty = 0$, (c) $\rho_\infty = 0.7$,$\rho^s_\infty = 0.5$, and (d) $\rho_\infty = \rho^s_\infty = 0$ (i.e., the L-stable algorithm). For this purpose, we employ a smaller Δt of 0.5s, and the results are

compared to the exact solution (equation (5.1.3)). Figures 5.8 and 5.9 show the solutions (y and $\dot{y}$ respectively) as a function of time. It is worthy to note that all plots in these figures show excellent agreement with the exact solution when the time step size is small enough.

Remark 5.1.4

1. *All algorithms in the GS4-1 framework yield excellent approximations to the exact solution when the time step size is small enough (see Figures 5.8 and 5.9).*

2. *More importantly, when larger time step sizes are used, the algorithms yield significant improvements by minimizing the numerical oscillations via the controllable numerical dissipation feature with the new selective control, while preserving other desirable attributes including second order accuracy, unconditional stability, and zero-order overshoot behaviour.*

We next show and prove that the algorithms within the newly developed GS4-1 framework are second order accurate. For this purpose, we show the convergence plots of the L-stable algorithm contained in our GS4-1 framework, namely GS4-1 with $(\rho_\infty,\rho_\infty^s) = (0,0)$, and the convergence plots of GS4-1 algorithm with arbitrarily chosen value of $(\rho_\infty,\rho_\infty^s) = (0.7,0.5)$. Figures 5.10 and 5.11 show the convergence plots for both variables generated by these algorithms. Note that the slope values shown in these figures represent the convergence rate (i.e., the order of accuracy) of the algorithms. For each algorithm, the total numbers of time steps (N_i, see equation (4.5.45)) used to construct the plots are chosen as 500000, 5000, 500, and 50, where the solutions generated with $N_i = 500000$ are used as the reference to calculate the errors for the construction of the convergence plots. Meanwhile, the end time at which the solutions are evaluated is $t_{end} = 5$s. We demonstrate the effect of the shift in $\dot{y}$ time level (as discussed in Section 4.5) by constructing: (a) the standard convergence plot for both y and $\dot{y}$, (b) the time level aligned convergence plot for both y and $\dot{y}$, and (c) the standard convergence plot for y and the time level aligned convergence plot for $\dot{y}$. As discussed earlier, the standard convergence plot does not take into account any shift in the time level. Since y does not have any time level shift, the standard convergence plot yields second order accuracy for y as expected. However, it yields only first order accuracy for $\dot{y}$ because of the shift in $\dot{y}$ time level as depicted in Figures 5.10(a) and 5.11(a). On the other hand, the time

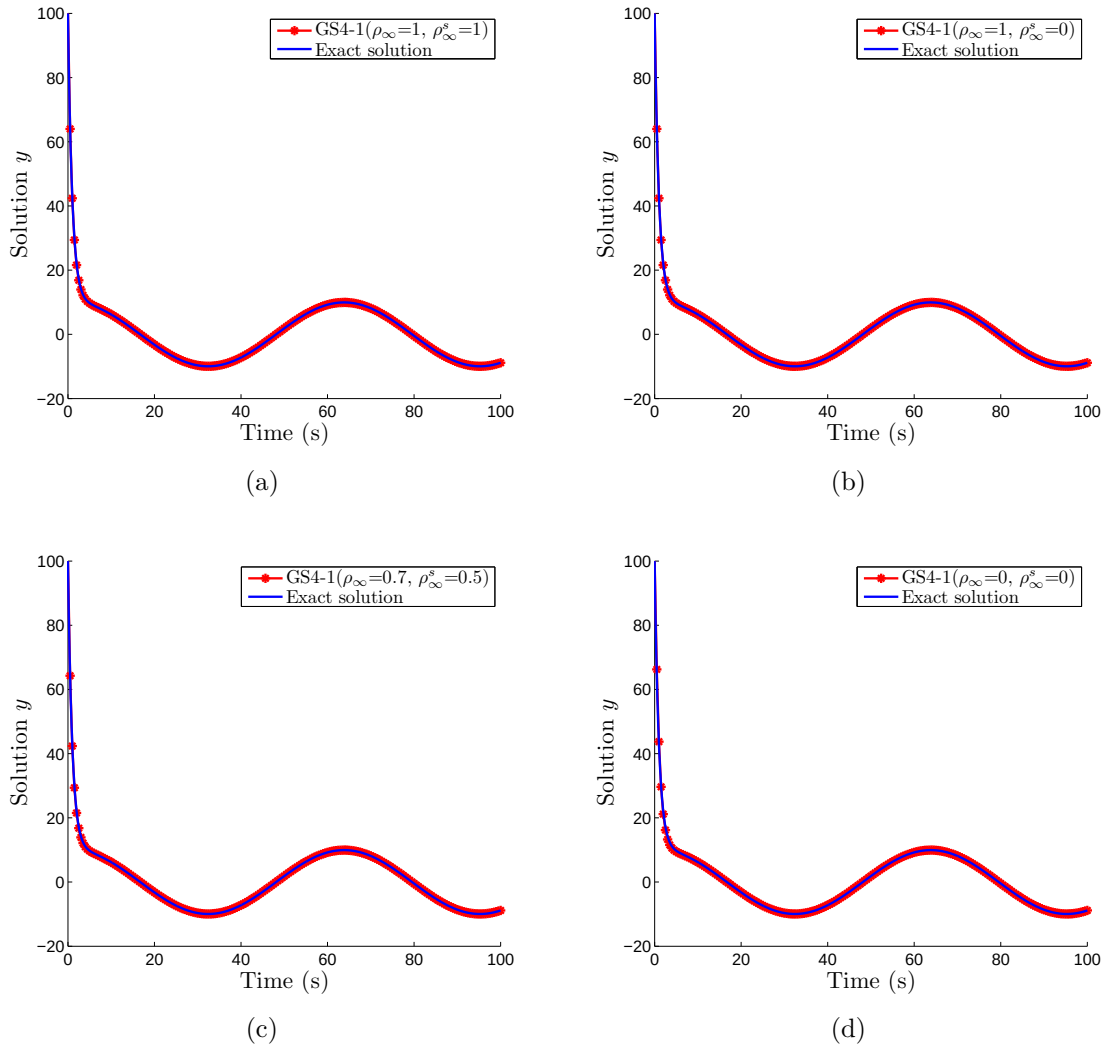


Figure 5.8: y (for the SDOF Example, section 5.1) as a function of time solved using: (a) Crank-Nicolson method / GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.5)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0,0)$, all of which solved using a time-step size $\Delta t = 0.5$s and are compared to the exact solution given by equation (5.1.3)

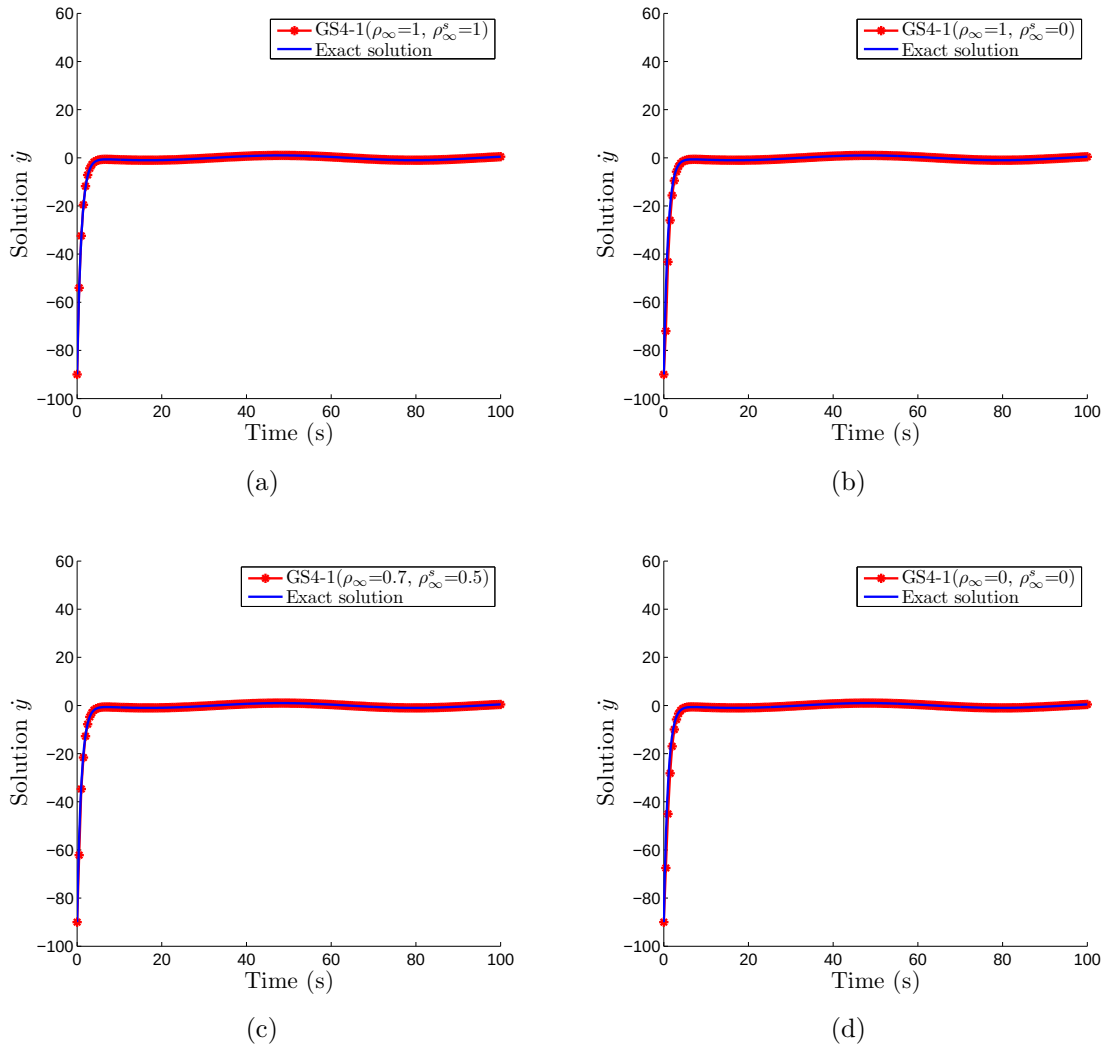
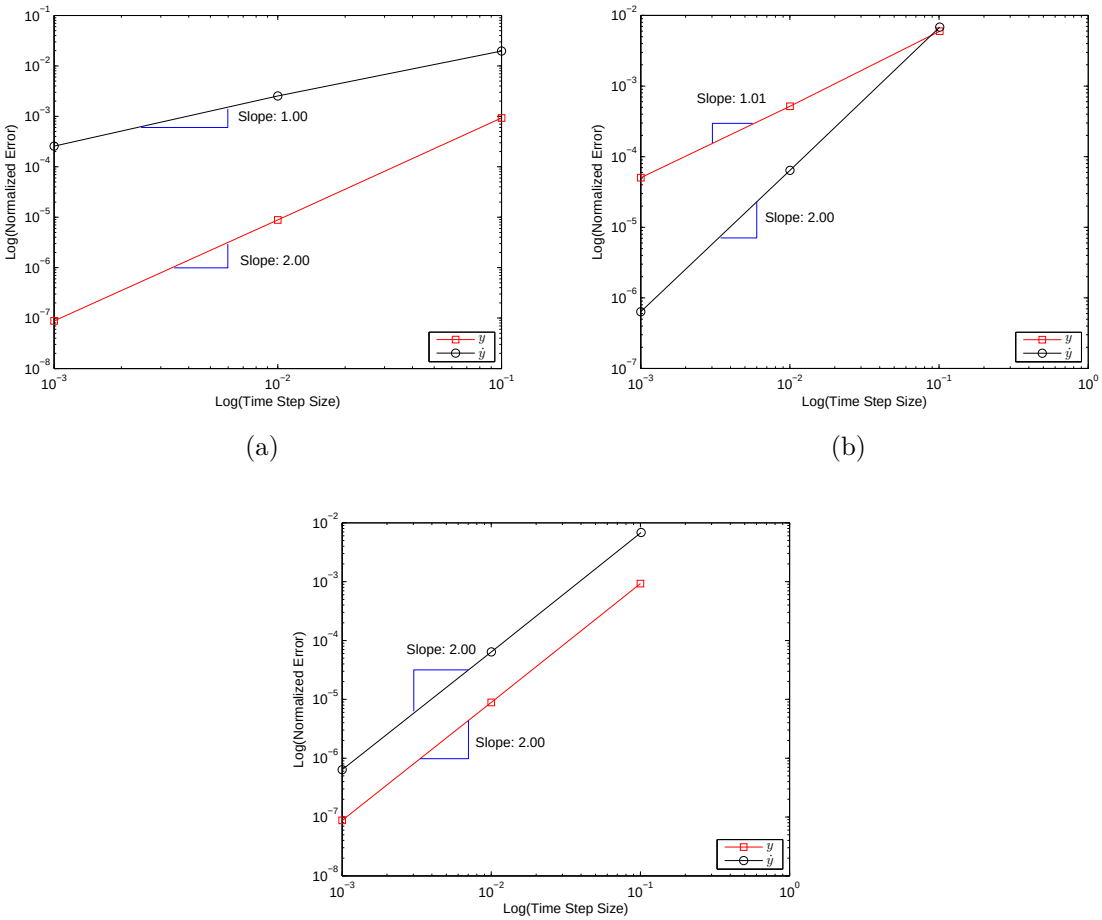


Figure 5.9: $\dot{y}$ (for the SDOF Example, section 5.1) as a function of time solved using: (a) Crank-Nicolson method / GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.5)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0,0)$, all of which solved using a time-step size $\Delta t = 0.5$s and are compared to the exact solution given by equation (5.1.3)

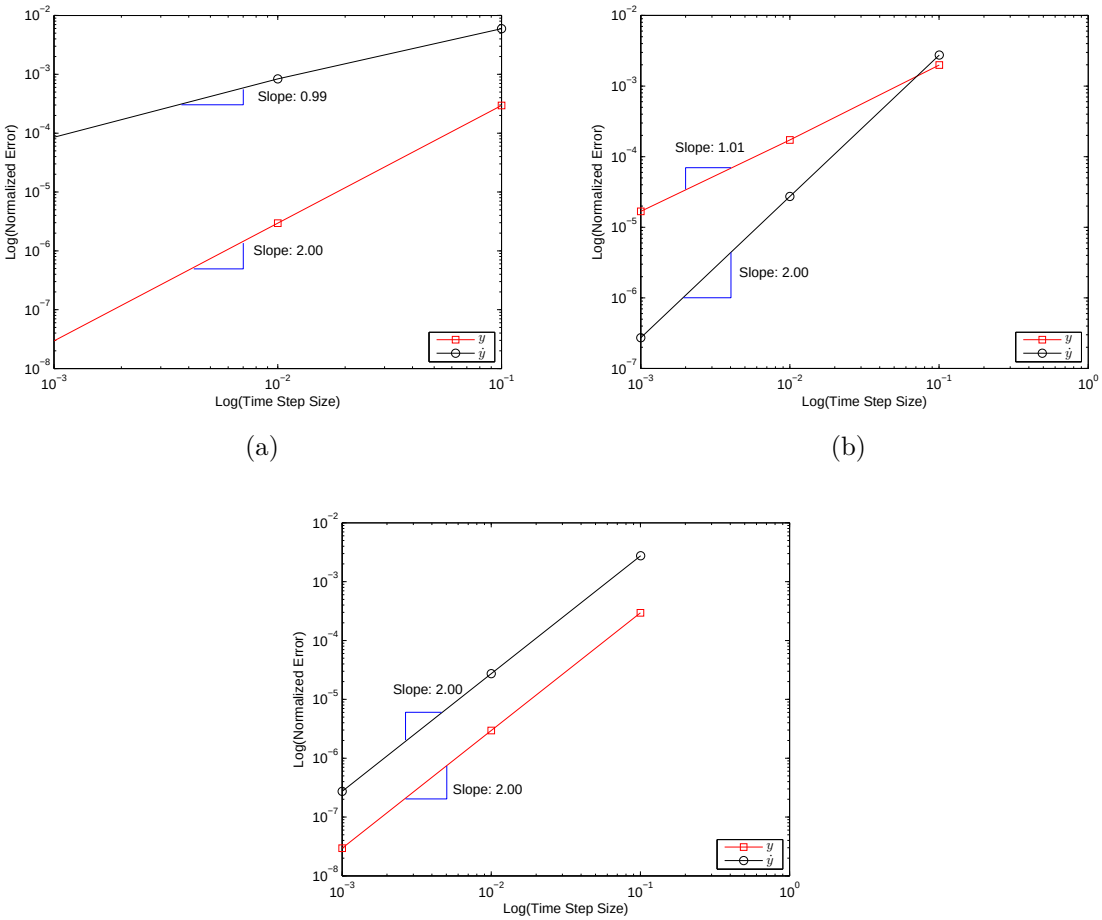
level aligned convergence plot takes into account the shift in time level. Therefore, it readily yields second order accuracy for $\dot{y}$ whereas y is only first order accurate (the correct alignment of $\dot{y}$ time level also means that y time level is not correctly aligned) as shown in Figures 5.10(b) and 5.11(b). Meanwhile, Figures 5.10(c) and 5.11(c) show that one could get second order accuracy for both y and $\dot{y}$ provided that the time level for each of these variables is correctly aligned, and it can be achieved by constructing the standard convergence plot for y and the time level aligned convergence plot for $\dot{y}$.



(a) (b)

(c)

Figure 5.10: Convergence plot of the solution of SDOF Example (section 5.1) solved using the developed GS4-1 framework with $(\rho_\infty,\rho_\infty^s) = (0,0)$ utilizing: (a) the standard convergence plot for both y and $\dot{y}$, (b) the time level aligned convergence plot for both y and $\dot{y}$, and (c) the standard convergence plot for y and the time level aligned convergence plot for $\dot{y}$



(a) (b)

(c)

Figure 5.11: Convergence plot of the solution of SDOF Example (section 5.1) solved using the developed GS4-1 framework with $(\rho_\infty,\rho^s_\infty) = (0.7,0.5)$ utilizing: (a) the standard convergence plot for both y and $\dot{y}$, (b) the time level aligned convergence plot for both y and $\dot{y}$, and (c) the standard convergence plot for y and the time level aligned convergence plot for $\dot{y}$

5.2 Parabolic Heat Conduction

The heat conduction problems describe the transfer of thermal energy between regions of an object due to differences in temperature. The problem could become time-dependent if given an imposed change in temperature at the object's boundary, or when a source (or sink) of heat is suddenly applied within the object and subsequently causes change in the nearby temperatures. In heat conduction problems, the analyst seeks to know the temperature distribution throughout the object from which all other secondary quantities can be determined including the rate of transfer of the thermal energy across an area (called the flux).

5.2.1 Governing Equation

For materials with constant specific heat capacity (c), density (ρ) and conductivity (k), the temperature distribution within the object, in the absence of sink/source term, is described by the following partial differential equation

$$\rho c\, \dot{T} = k \nabla^2 T \tag{5.2.4}$$

where $T = T(x, y, z, t)$ represents the temperature at point (x, y, z) and time t.

Following reference [2], we consider in this section a two-dimensional rectangular slab with initial temperature of unity which is uniform over the entire domain. One the left boundary ($x = 0$), the Dirichlet boundary condition for the temperature is set to zero (cooled side) while all other boundaries are insulated (null heat flux) as depicted in Figure 5.12(a). The physical properties of the material are: $\rho = 1.0\ kg/m^3$, $c = 1.0\ J/(kg\ K)$ and $k = 1.0\ W/(mK)$.

5.2.2 Spatial Discretization by the Finite Element Method

Applying the method of weighted residuals to discretize the governing equation (equation (5.2.4)), we choose m linearly independent weighting functions W_i to satisfy

$$\int_\Omega W_i(\rho c\, \dot{T} - k \nabla^2 T) \partial\Omega = 0, \quad i = 1, 2, ..., m \tag{5.2.5}$$

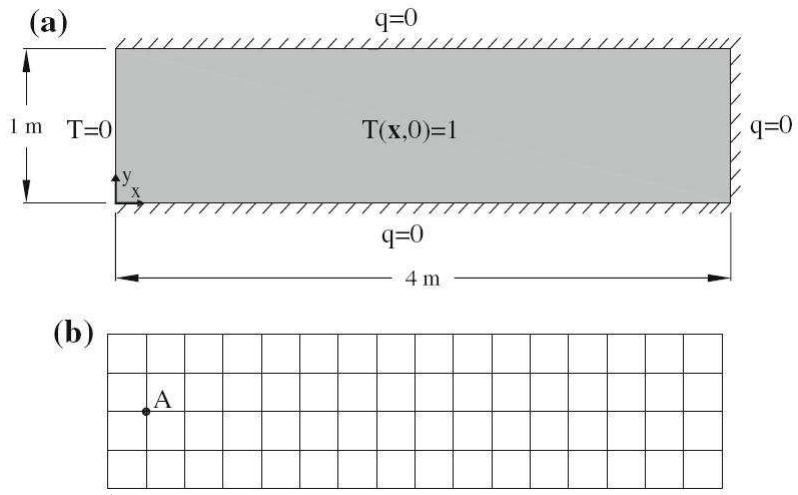


Figure 5.12: 2D rectangular slab [2]: (a) geometry and boundary conditions; (b) finite element mesh and the selected node

where Ω is the entire solution domain. However, since equation (5.2.5) holds for any point in the solution domain, it also holds for any collection of points defining an arbitrary element within the whole domain. Consequently, equation (5.2.5) can be written for an individual element (e) having r nodes as follows

$$\int_{\Omega^{(e)}} W_i^{(e)}(\rho c\, \dot{T}^{(e)} - k\nabla^2 T^{(e)})\partial\Omega = 0, \quad i = 1, 2, ..., r \tag{5.2.6}$$

where $\Omega^{(e)}$ represents the domain of the element. By the usual procedure, we approximate the temperature within the element as follows

$$T^{(e)}(x, y, t) = \sum_{i=1}^{r} N_i^{(e)}(x, y) T_i^{(e)}(t) \tag{5.2.7}$$

where $N_i^{(e)}(x, y)$ is the shape function for the element and $T_i^{(e)}(t)$ is the nodal temperature. Applying the Green-Gauss theorem to the diffusion term in equation (5.2.6), and substituting the approximation (equation (5.2.7)) into equation (5.2.6), we have the following system of first order (in time) ordinary differential equations upon imposing the null heat flux boundary conditions

$$\mathbf{M}^{(e)}\dot{\mathbf{T}}^{(e)} + \mathbf{K}^{(e)}\mathbf{T}^{(e)} = \mathbf{F}^{(e)} \tag{5.2.8}$$

where

$$M_{ij}^{(e)} = \rho c \int_{\Omega^{(e)}} W_i^{(e)} N_j^{(e)} \partial\Omega \tag{5.2.9}$$

$$K_{ij}^{(e)} = k \int_{\Omega^{(e)}} \nabla W_i^{(e)} \nabla N_j^{(e)} \partial\Omega \tag{5.2.10}$$

$$F_i^{(e)} = 0 \tag{5.2.11}$$

There are various weighted residual techniques giving different forms of the weighting function $W_i^{(e)}$. For this example, we use the Bubnov-Galerkin method (i.e., the standard Galerkin method), in which the weighting functions are chosen to be the same as the shape functions used to approximate the variable, that is $W_i^{(e)} = N_i^{(e)}$. The entire solution domain Ω is divided into 64 quadrilateral elements as shown in Figure 5.12(b). Upon the assembly of all elements, we obtain the following global system of equations

$$\mathbf{M}\dot{\mathbf{T}} + \mathbf{K}\mathbf{T} = \mathbf{F} \tag{5.2.12}$$

where

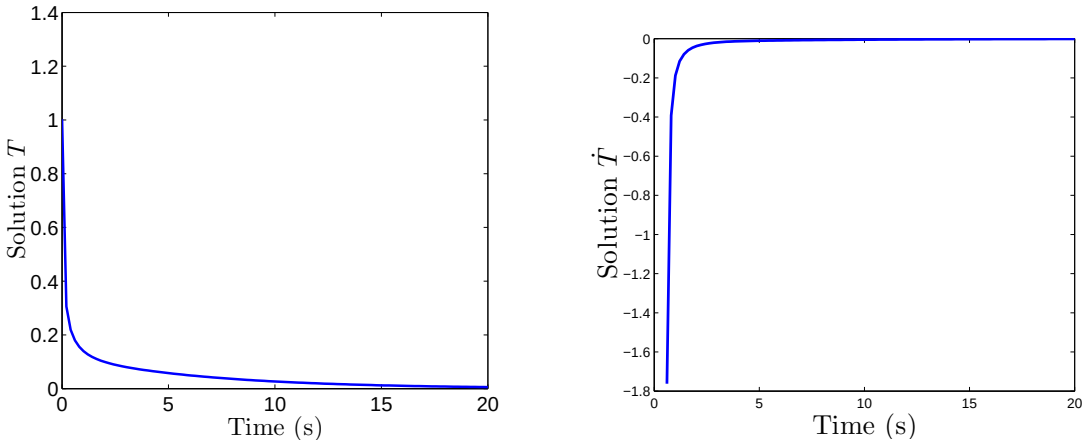
$$\mathbf{M} = \sum_{1}^{64} \mathbf{M}^{(e)} \tag{5.2.13}$$

$$\mathbf{K} = \sum_{1}^{64} \mathbf{K}^{(e)} \tag{5.2.14}$$

$$\mathbf{F} = 0 \tag{5.2.15}$$

5.2.3 Time Integration by the GS4-1 Framework

To solve equation (5.2.12), we employ the developed GS4-1 framework to march the solutions from $t_0 = 0s$ to $t_{end} = 20s$ with a time step size of $\Delta t = 1s$. For the illustration of the numerical results, we consider a point A defined by $x = 0.25m$ and $y = 0.5m$ as shown in Figure 5.12(b). The corresponding analytical solutions of the primary variable and its time derivative as a function of time are given in [2] and shown in Figure 5.13. For this problem, we repeat the same analysis as done in the SDOF example (Section 5.1), i.e.,: (1) the results of the noteworthy family of GS4-1 algorithms with $\rho_\infty = 1$ (*Algorithm* 4), (2) the roles of the two parameters (ρ_∞ and ρ_∞^s), and (3) the results of the L-stable algorithm, to show consistency of such attributes of the developed GS4-1 framework for this Multi-Degree-of-Freedom (MDOF) problem as those seen in the SDOF example. The relevant results are presented in Figures 5.14 to 5.20. It is worthy to note from these numerical results that similar remarks as those in the SDOF example can be drawn. Therefore, the consistency of the argument is proven. Such remarks are not presented in this example to avoid unnecessary repetition.



(a) (b)

Figure 5.13: Plot of analytical solutions T and $\dot{T}$ (for the parabolic heat conduction problem, Section 5.2) as a function of time for node A (see Figure 5.12) as given in [2]

Meanwhile, Figures 5.21 and 5.22 show the convergence plots of the solutions of node

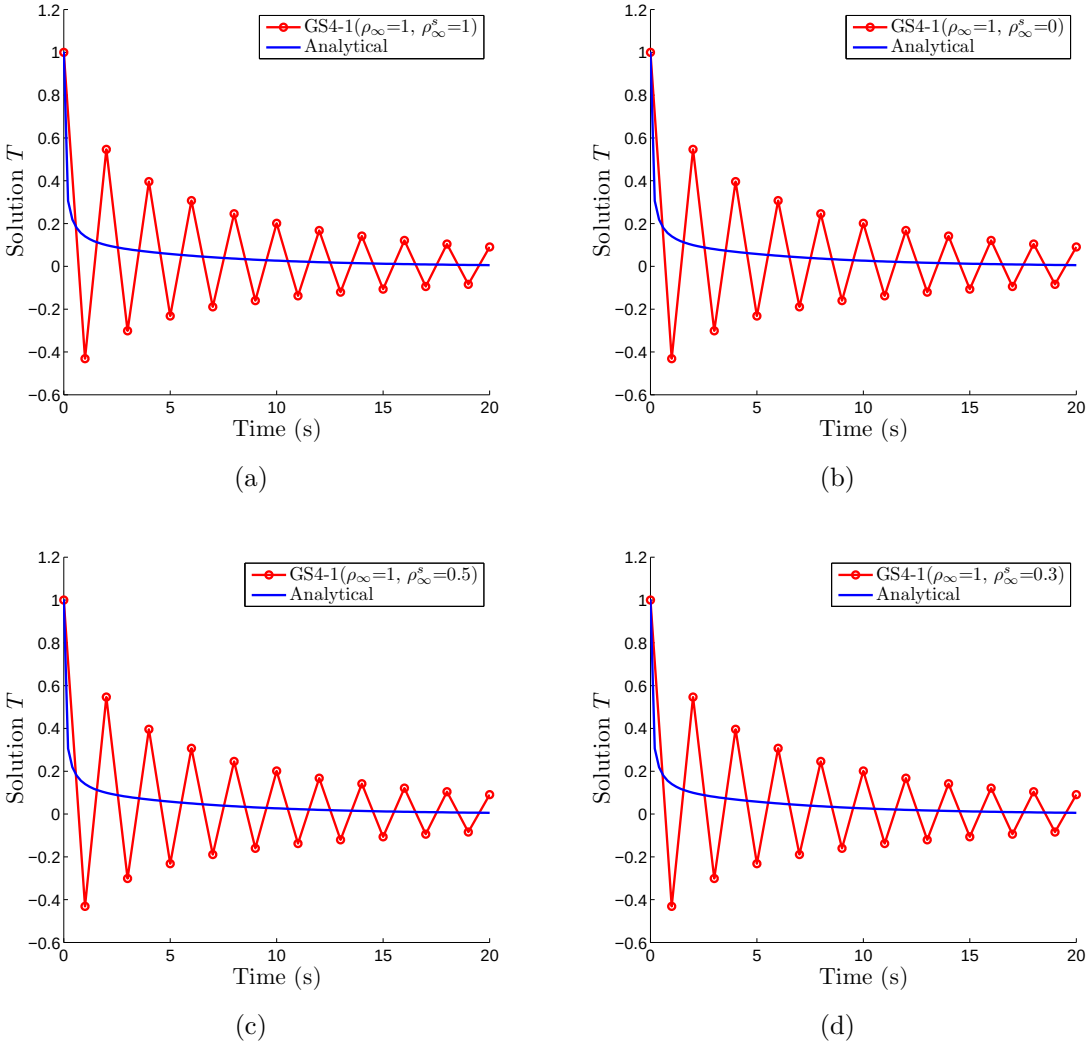


Figure 5.14: T (for the parabolic heat conduction problem, Section 5.2) as a function of time solved using: (a) Crank-Nicolson method / GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0.5)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (1,0.3)$

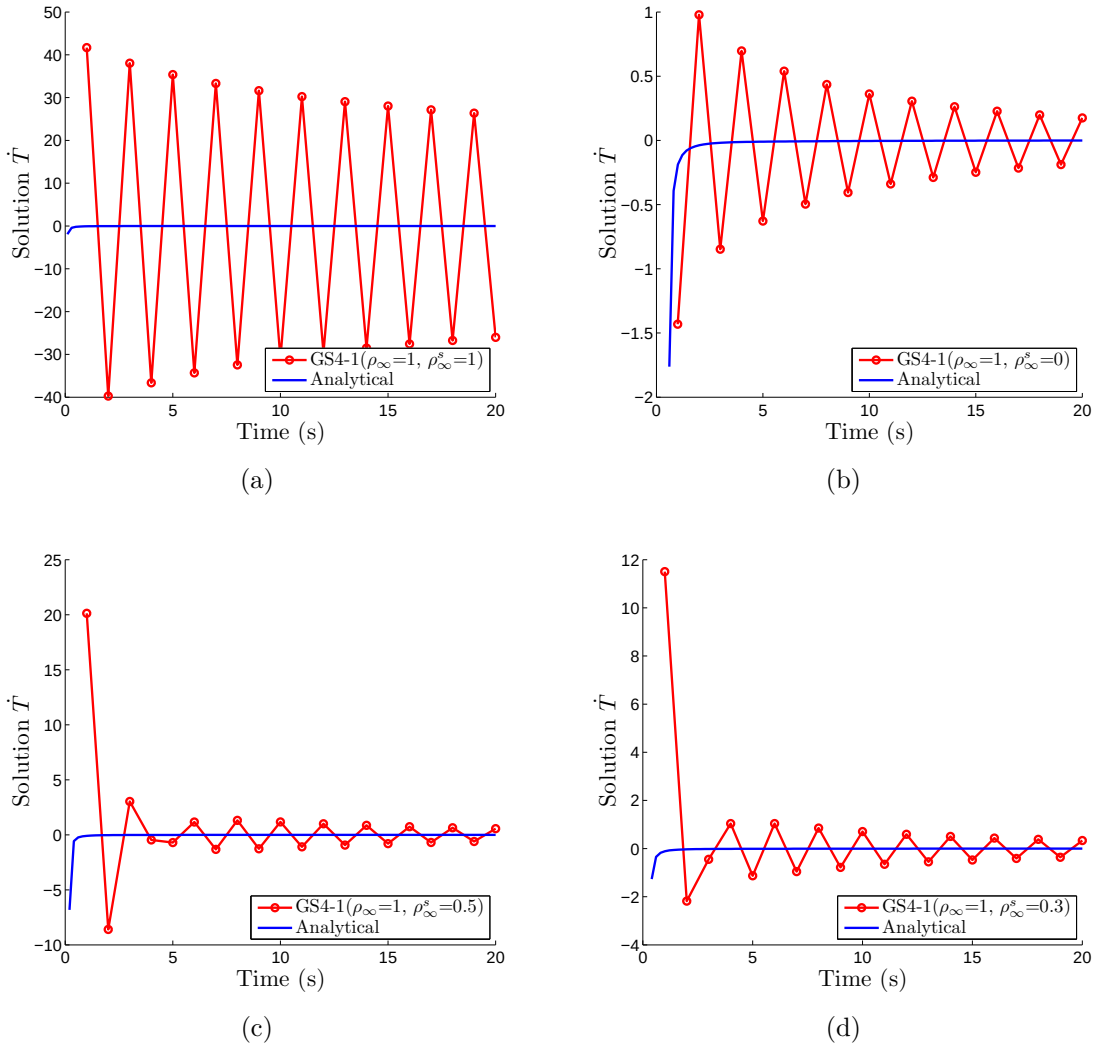


Figure 5.15: $\dot{T}$ (for the parabolic heat conduction problem, Section 5.2) as a function of time solved using: (a) Crank-Nicolson method / GS4-1 with (ρ_∞,ρ^s_∞) = (1,1), (b) GS4-1 with (ρ_∞,ρ^s_∞) = (1,0), (c) GS4-1 with (ρ_∞,ρ^s_∞) = (1,0.5) and (d) GS4-1 with (ρ_∞,ρ^s_∞) = (1,0.3)

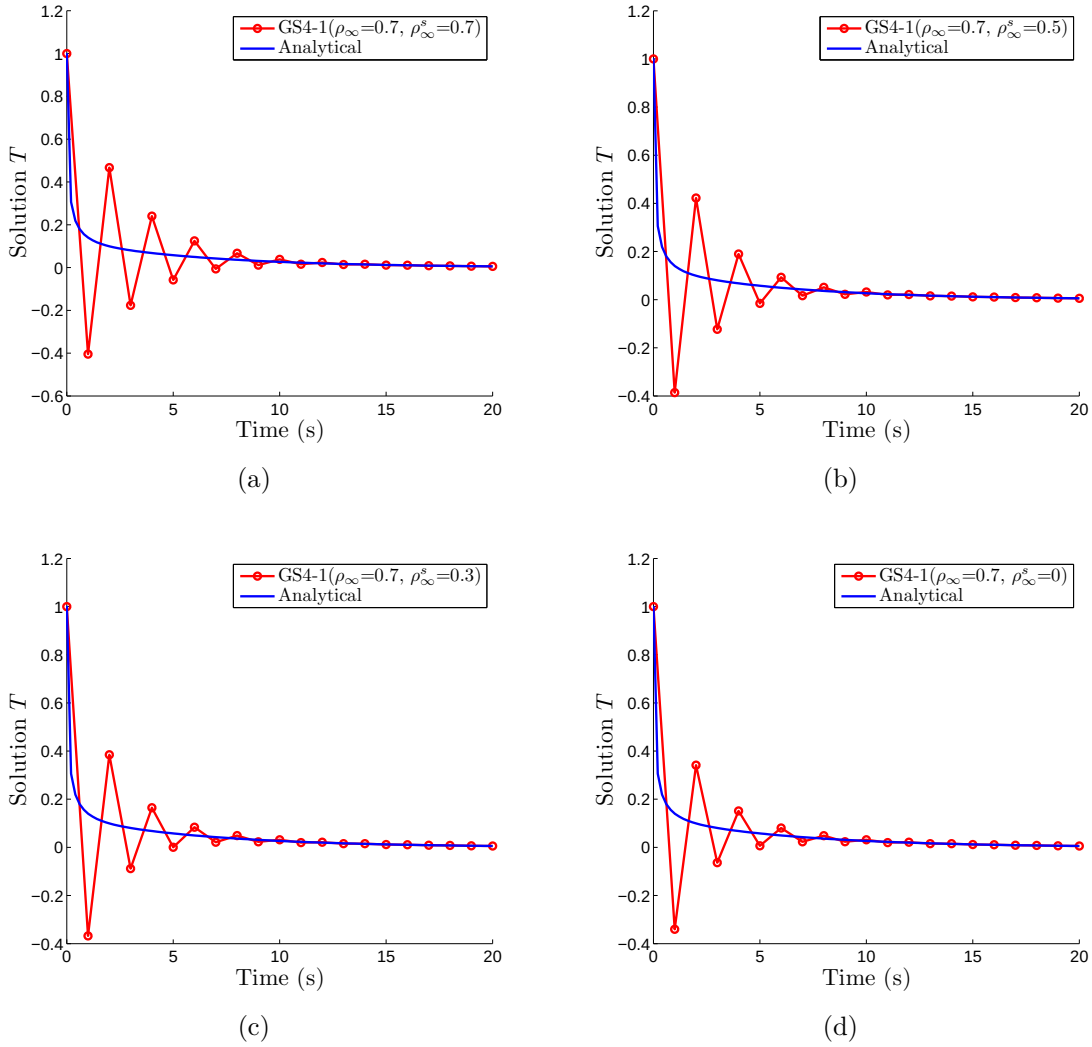


Figure 5.16: T (for the parabolic heat conduction problem, Section 5.2) as a function of time solved using: (a) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.7), (b) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.5), (c) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.3) and (d) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.0)

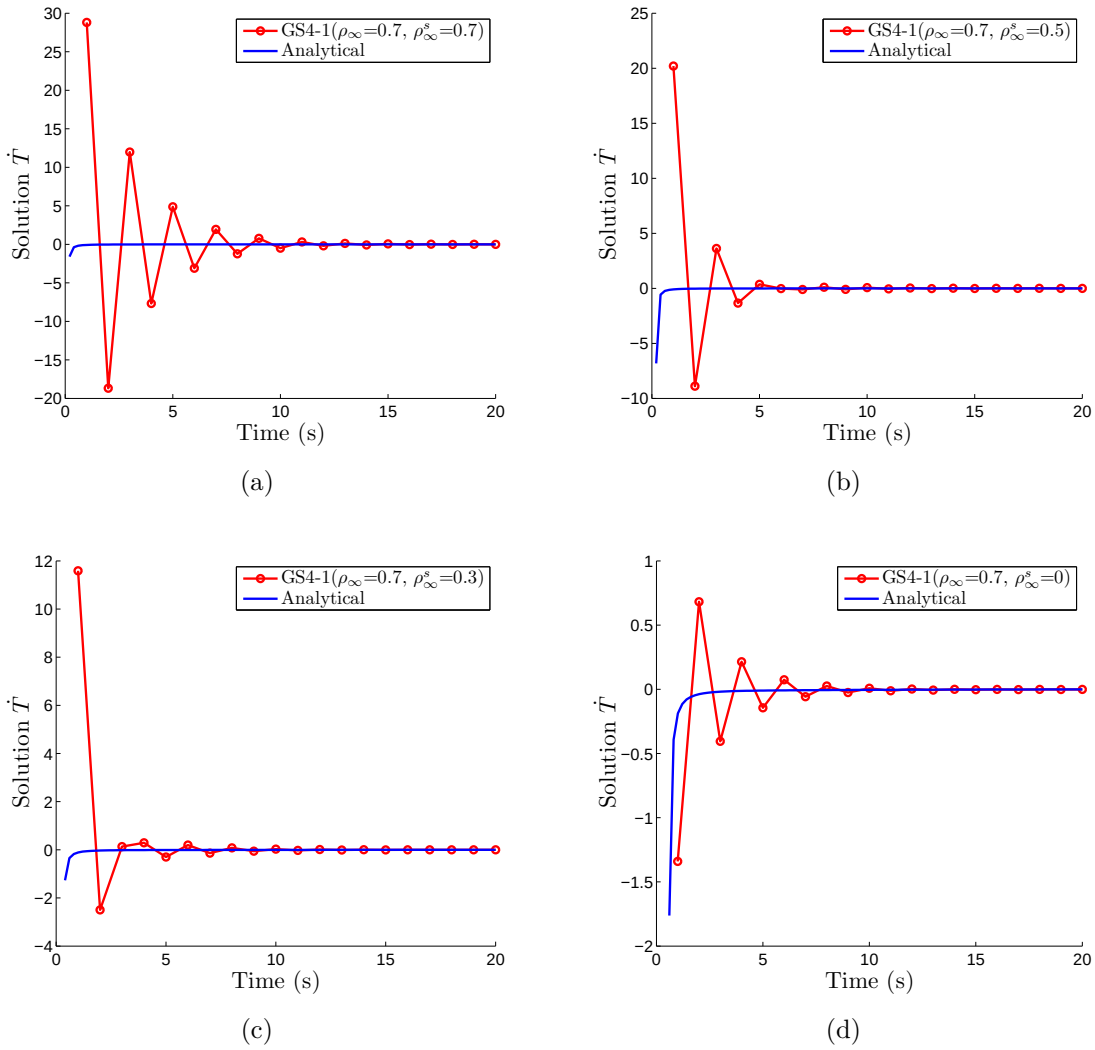


Figure 5.17: $\dot{T}$ (for the parabolic heat conduction problem, Section 5.2) as a function of time solved using: (a) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.7), (b) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.5), (c) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.3) and (d) GS4-1 with (ρ_∞,ρ^s_∞) = (0.7,0.0)

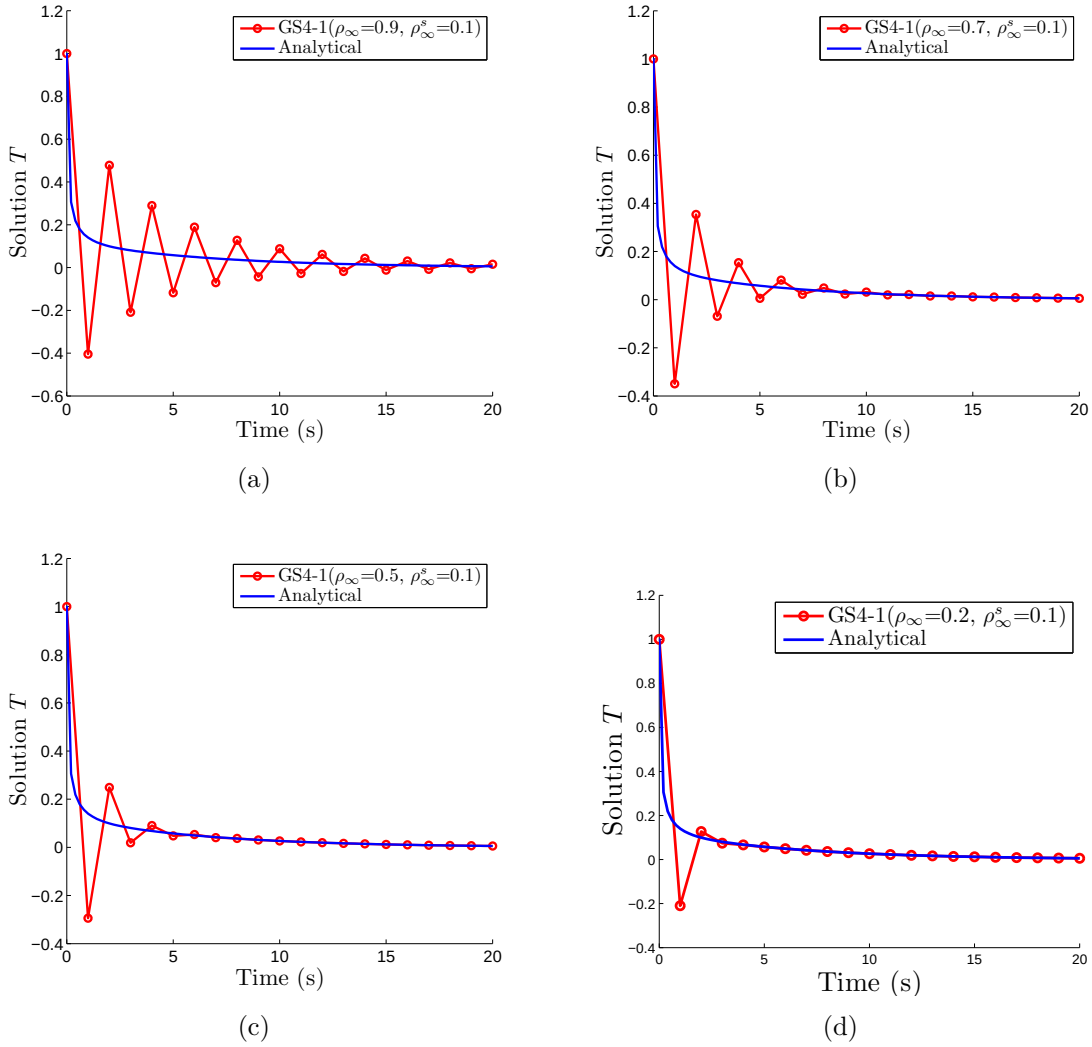


Figure 5.18: T (for the parabolic heat conduction problem, Section 5.2) as a function of time solved using: (a) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.9,0.1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.1)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.5,0.1)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.2,0.1)$

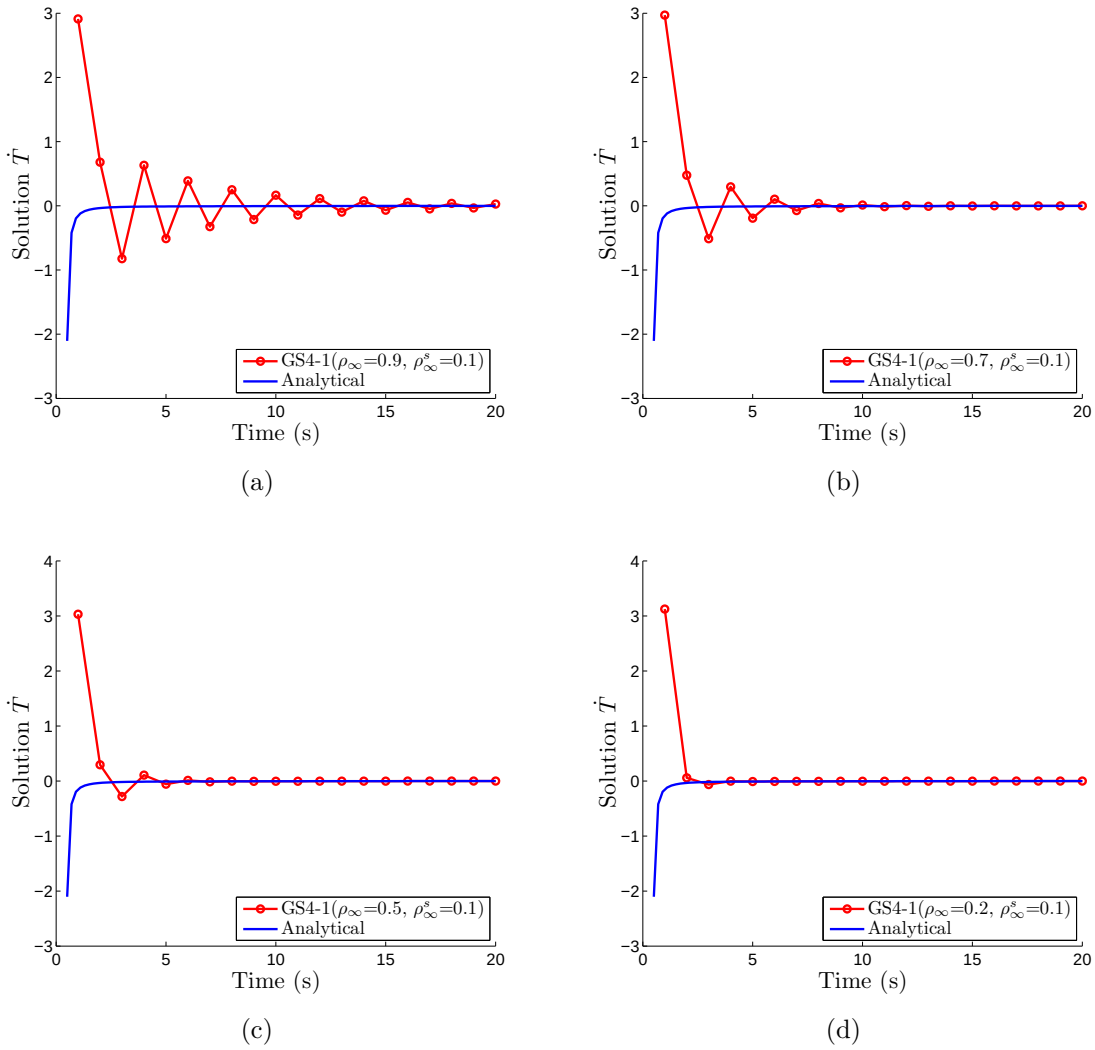
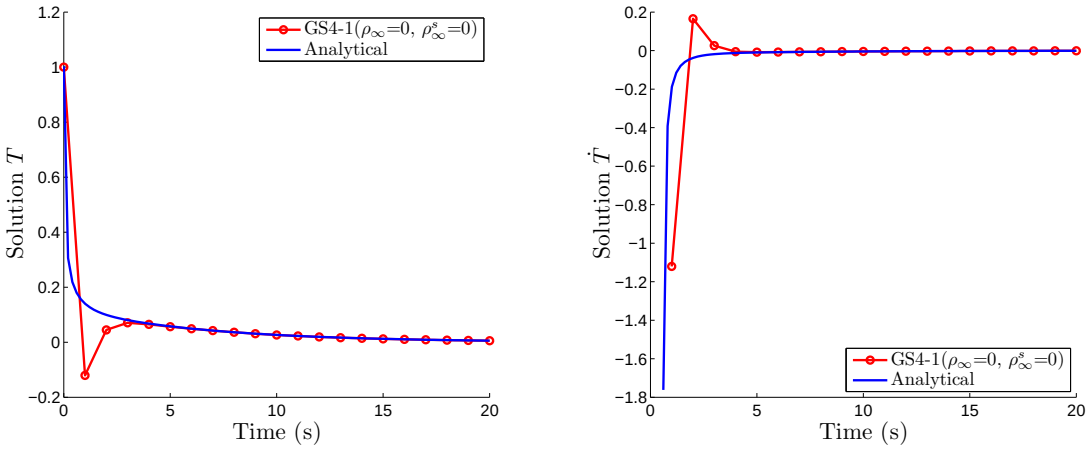


Figure 5.19: $\dot{T}$ (for the parabolic heat conduction problem, Section 5.2) as a function of time solved using: (a) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.9,0.1)$, (b) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.7,0.1)$, (c) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.5,0.1)$ and (d) GS4-1 with $(\rho_\infty,\rho^s_\infty) = (0.2,0.1)$



(a) (b)

Figure 5.20: The solutions T and $\dot{T}$ of the parabolic heat conduction problem (section 5.2) as a function of time solved using GS4-1 with (ρ_∞,ρ^s_∞) = (0,0)

A solved using the developed GS4-1 framework with (ρ_∞,ρ^s_∞) = (0.7,0.5) and (ρ_∞,ρ^s_∞) = (0,0), respectively, which demonstrate the order of accuracy of these algorithms. The total number of time steps (N_i, see equation (4.5.45)) used to construct the plots are chosen as 500000, 5000, 500, and 50, where the solutions generated with N_i = 500000 are used as the reference to calculate the errors for the construction of the convergence plots. Meanwhile, the end time at which the solutions are evaluated is t_{end} = 5s. As done in the SDOF example (section 5.1), we demonstrate in each figure the effect of shift in $\dot{T}$ time level by constructing: (a) the standard convergence plot for both T and $\dot{T}$, (b) the time level aligned convergence plot for both T and $\dot{T}$, and (c) the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$. Similar observations and conclusions about the convergence rate as those seen in the SDOF example (Figures 5.10 and 5.11) can be seen in Figures 5.21 and 5.22, and hence demonstrate the consistency.

We now proceed to illustrate using this numerical example the effect of the new selective control feature of the newly developed GS4-1 framework. For this purpose, we solve the problem using the two cases under investigation (see *Remark* 4.3.1) i.e.,: (1) the GS4-1 framework without the selective control feature which recovers the existing method [62], and (2) the GS4-1 framework with the new selective control feature, with a time step size (Δt) of 1s and an end time of 20s. For illustration

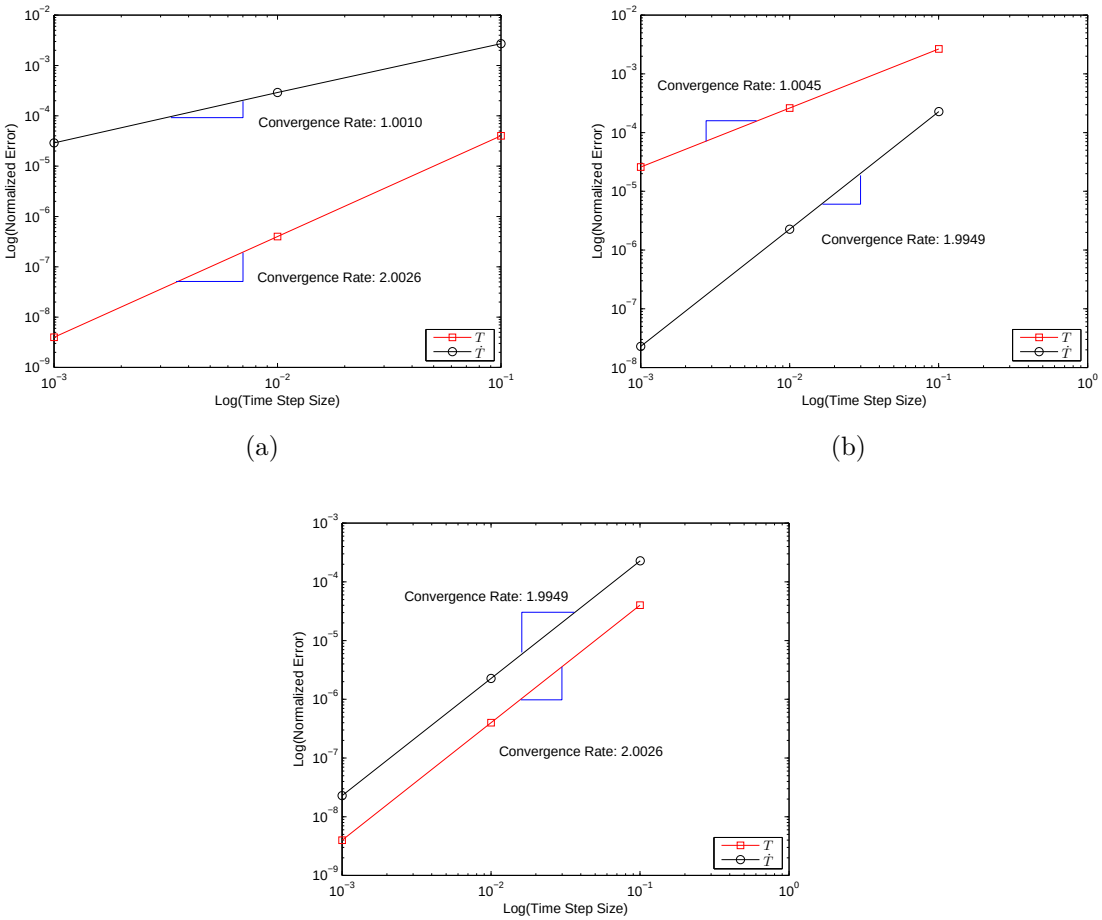
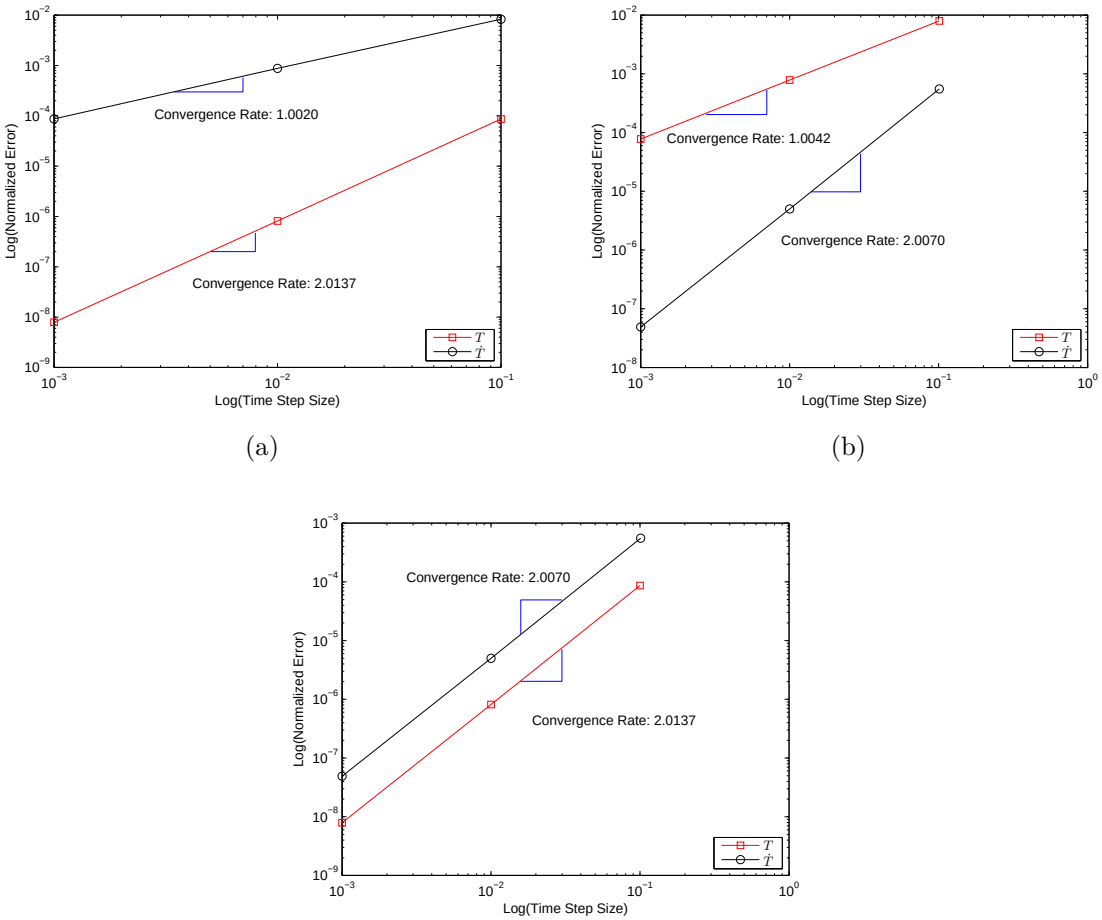


Figure 5.21: Convergence plot of the solutions of node A (for the parabolic heat conduction problem, Section 5.2) solved using the developed GS4-1 framework with $(\rho_\infty,\rho^s_\infty) = (0.7,0.5)$ utilizing: (a) the standard convergence plot for both T and $\dot{T}$, (b) the time level aligned convergence plot for both T and $\dot{T}$, and (c) the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$



(a) (b)

(c)

Figure 5.22: Convergence plot of the solutions of node A (for the parabolic heat conduction problem, Section 5.2) solved using the developed GS4-1 framework with $(\rho_\infty,\rho^s_\infty) = (0,0)$ utilizing: (a) the standard convergence plot for both T and $\dot{T}$, (b) the time level aligned convergence plot for both T and $\dot{T}$, and (c) the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$

purpose, we compare the performance of the two cases for a randomly chosen ρ_∞ value of 0.4. The comparison of the performance between these cases is done by first looking at the solutions of T and $\dot{T}$ as a function of time for node A. Recall that the analytical solutions, as given in [2], are shown in Figure 5.13. Meanwhile, the numerical solutions of T and $\dot{T}$ as a function of time for this node generated by the two cases are shown in Figure 5.23. An observation of this figure reveals the improvement in the numerical solutions of both T and $\dot{T}$ made by the case with the selective control feature (i.e., the GS4-1 algorithm with $\rho_\infty = 0.4$ and $\rho_\infty^s = 0$), in contrast to the case without such a feature (i.e., $\rho_\infty = \rho_\infty^s = 0.4$) for the same value of the principal root (i.e., $\rho_\infty = 0.4$). This difference in performance between the two cases highlights the role played by the new selective control feature of the developed GS4-1 framework.

For a complete investigation, we then compute and compare the errors in both T and $\dot{T}$ generated by these two cases for ρ_∞ values ranging from 1 (nondissipative/zero damping) to 0 (maximal damping) in decrements of 0.1. The error is defined as

$$Error = |Numerical - Analytical| \tag{5.2.16}$$

Tables 5.2.1 and 5.2.2 compare the maximal and total errors in the solutions of T and $\dot{T}$, respectively, as generated by the two cases (i.e., the GS4-1 framework with and without the selective control feature) for all ρ_∞ values considered. By looking at Table 5.2.1, it is obvious that the errors in T generated by the two cases are very small. In other words, we can say that both cases perform well to obtain acceptable solutions of T. On the other hand, Table 5.2.2 shows that the errors in $\dot{T}$ (both maximal and total) generated by the case without the selective control feature are large except for large amount of numerical dissipation ($0 \leq \rho_\infty \leq 0.6$) which is undesirable (due to possible over-dissipation). Interestingly, these errors are significantly reduced by the GS4-1 framework when the selective control feature is enabled. Therefore, we conclude that the GS4-1 framework with the selective control feature provides an improvement over the case without such a feature as in past development. This will be demonstrated in the following numerical examples throughout the thesis to prove consistency of the conclusion.

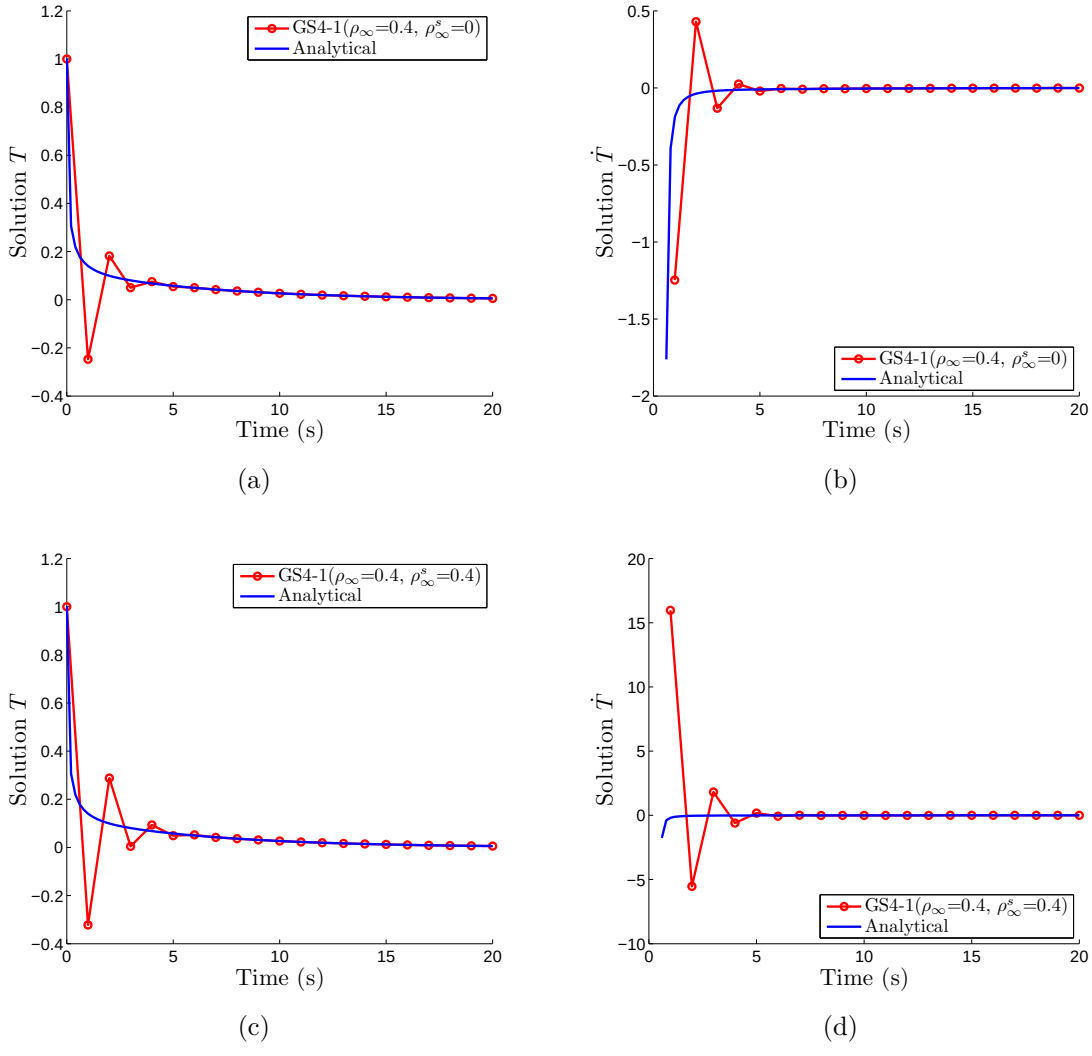


Figure 5.23: Plot of T and $\dot{T}$ as a function of time for node A (for the parabolic heat conduction problem, Section 5.2) generated by: (i) GS4-1($\rho_\infty = 0.4, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 0.4, \rho^s_\infty = 0.4$) i.e., the case without the selective control feature

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	0.0850	0.0850	0.3275	0.3275
0.9	0.0203	0.0101	0.1062	0.0479
0.8	0.0026	$8.6609\text{x}10^{-4}$	0.0217	0.0156
0.7	$5.6841\text{x}10^{-4}$	$6.8083\text{x}10^{-4}$	0.0117	0.0138
0.6	$5.9327\text{x}10^{-4}$	$7.3724\text{x}10^{-4}$	0.0121	0.0150
0.5	$6.2957\text{x}10^{-4}$	$8.0159\text{x}10^{-4}$	0.0128	0.0163
0.4	$6.8459\text{x}10^{-4}$	$8.7569\text{x}10^{-4}$	0.0139	0.0178
0.3	$7.6622\text{x}10^{-4}$	$9.6192\text{x}10^{-4}$	0.0156	0.0196
0.2	$8.8666\text{x}10^{-4}$	0.0011	0.0180	0.0216
0.1	0.0011	0.0012	0.0217	0.0241
0	0.0013	0.0013	0.0271	0.0271

Table 5.2.1: Comparison of error in T (for the parabolic heat conduction problem, Section 5.2) between: (i) the case without the selective control feature (SCF)(i.e., GS4-1 framework with $\rho_\infty^s = \rho_\infty$), and (ii) the case with the selective control feature (i.e., GS4-1 framework with $\rho_\infty^s = 0$), for ρ_∞ values ranging from 1 (nondissipative/zero damping) to 0 (maximal damping) in decrements of 0.1.

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	70.2769	0.1757	226.9521	0.6587
0.9	8.5440	0.0222	24.6318	0.0843
0.8	0.8102	0.0022	48.7307	0.0091
0.7	0.0561	$1.5954\text{x}10^{-4}$	0.1437	0.0018
0.6	0.0026	$6.7106\text{x}10^{-5}$	0.0082	0.0014
0.5	8.236610^{-5}	$7.1950\text{x}10^{-5}$	0.0018	0.0015
0.4	$8.5200\text{x}10^{-5}$	$7.7475\text{x}10^{-5}$	0.0017	0.0016
0.3	$8.8950\text{x}10^{-5}$	$8.3826\text{x}10^{-5}$	0.0018	0.0017
0.2	$9.4064\text{x}10^{-5}$	$9.1198\text{x}10^{-5}$	0.0019	0.0019
0.1	$1.0097\text{x}10^{-4}$	$9.9841\text{x}10^{-5}$	0.0021	0.0020
0	$1.1008\text{x}10^{-4}$	$1.1008\text{x}10^{-4}$	0.0022	0.0022

Table 5.2.2: Comparison of error in $\dot{T}$ (for the parabolic heat conduction problem, Section 5.2) between: (i) the case without the selective control feature (SCF)(i.e., GS4-1 framework with $\rho_\infty^s = \rho_\infty$), and (ii) the case with the selective control feature (i.e., GS4-1 framework with $\rho_\infty^s = 0$), for ρ_∞ values ranging from 1 (nondissipative/zero damping) to 0 (maximal damping) in decrements of 0.1.

5.3 Flow Transport Problems

Flow transport is the movement of a constituent (mass, energy, a component in a mixture, solid particles, etc) of a carrying fluid due to the combination of the forces acting on the constituent and the movement of the carrying fluid. Typical applications of this phenomenon include the deposition of fine solid particles from a liquid suspension onto the walls of the equipments encountered in industrial processes and manufacturing technologies. For instance, deposition mechanism is applied in the preparation of magnetic tapes and in purification of water using packed-bed filters. On the other hand, there is a great concern in geothermal power plant to reduce this mechanism, which commonly occurs due to the presence of silica or magnesium in the geothermal brine from which heat is transferred to the working fluid in a heat exchanger device. A better understanding and knowledge of these mechanisms can be used to tackle or improve practical design of these processes and devices. Flow transport also governs the transport of viruses and associated pollutants in soils and ground water, as well as sediment transport which occurs in rivers, oceans, lakes, seas, and other bodies of water due to currents and tides. Mass transport is common in adsorption and absorption processes such as in carbon filtration, in the separation of chemical components such as distillation columns, and in cooling towers.

5.3.1 Governing Equation

The equations describing these problems are usually linear in nature, and are composed of two components: (1) convective transport, and (2) diffusive transport. Therefore, such equations are usually known as the convection-diffusion equations due to the presence of these two components. The convective transport accounts for the transport of the constituent due to the existence of the velocity field. Meanwhile, the diffusive transport represents the random motions of the huge number of the constituent molecules (that make up the carrying fluid) from regions of higher concentration to regions of lower concentration as driven by the concentration gradient. In general, it is necessary to account for both components to describe flow transport problems. For such linear cases, the time-dependent convection-diffusion equation can be written in the following form (in the absence of the sink/source term)

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} + \mathbf{v}\cdot\nabla\phi(\mathbf{x},t) = \nabla\cdot\kappa\nabla\phi(\mathbf{x},t), \quad \forall \mathbf{x}\in\Omega\subset R^d, t>0 \tag{5.3.17}$$

with conditions on the boundary Γ as

$$\phi(\mathbf{x},t) = \phi_\Gamma(\mathbf{x},t) \quad \forall \mathbf{x}\in\Gamma_1 \tag{5.3.18}$$

$$\kappa\nabla\phi(\mathbf{x},t).\vec{\mathbf{n}} = q(\mathbf{x}) \quad \forall \mathbf{x}\in\Gamma_2 \tag{5.3.19}$$

$$\Gamma = \Gamma_1 + \Gamma_2 \tag{5.3.20}$$

and initial condition

$$\phi(\mathbf{x}, t=0) = \phi_0(\mathbf{x}) \quad \forall \mathbf{x}\in\Omega \tag{5.3.21}$$

where $\phi(\mathbf{x},t)$ is the primary variable representing the constituent concentration at position $\mathbf{x}$ and time t, $\mathbf{v} = (v_1, v_2, ..., v_d)$ is the velocity vector of the carrying fluid, κ is the diffusion coefficient tensor, $\mathbf{x} = (x_1, x_2, ..., x_d)$ is the position vector, d is the dimension of the problem, Ω is a bounded domain in R^d, Γ is the boundary of Ω, and $\phi_\Gamma(\mathbf{x},t)$, $q(\mathbf{x})$, and ϕ_0 are known boundary and initial conditions.

Equation (5.3.17) can be written in dimensionless form by introducing dimensionless primary variable $\hat{\phi} = \frac{\phi-\phi_{ref_1}}{\phi_{ref_2}-\phi_{ref_1}}$, velocity $\hat{v}_i = \frac{v_i}{U}$, time $\hat{t} = \frac{Ut}{L}$ and coordinate $\hat{x}_i = \frac{x_i}{L}$, where L is the specific length of the domain, U is the characteristic velocity of the flow, while ϕ_{ref_1} and ϕ_{ref_2} are the lower and upper reference values of the primary variables, respectively. From the use of these dimensionless variables, equation (5.3.17) can be represented in a dimensionless form as follows

$$\frac{\partial \hat{\phi}(\hat{\mathbf{x}},\hat{t})}{\partial \hat{t}} + \hat{\mathbf{v}}\cdot\nabla\hat{\phi}(\hat{\mathbf{x}},\hat{t}) = \frac{1}{Pe}\nabla^2\hat{\phi}(\hat{\mathbf{x}},\hat{t}) \tag{5.3.22}$$

where $Pe = \frac{UL}{\kappa}$ is the global Peclet number. Throughout this section, we shall consider this dimensionless governing equation (equation (5.3.22)), however with the superscript ($\hat{}$) omitted in the notations for simplicity.

5.3.2 Spatial Discretization by the Finite Element Method

To discretize the spatial domain using the Finite Element Method, we apply the method of weighted residuals to equation (5.3.22) for an arbitrary element (e) with r nodes

$$\int_{\Omega^{(e)}} W_i^{(e)} \left(\frac{\partial \phi^{(e)}(\mathbf{x},t)}{\partial t} + \mathbf{v} \cdot \nabla \phi^{(e)}(\mathbf{x},t) - \frac{1}{Pe} \nabla^2 \phi^{(e)}(\mathbf{x},t) \right) \partial\Omega = 0 \quad i = 1, 2, ..., r \tag{5.3.23}$$

where $W_i^{(e)}$ is the weighting function and $\Omega^{(e)}$ is the domain for the element. We next apply Gauss's theorem to the diffusive term as follows

$$\begin{aligned} \int_{\Omega^{(e)}} W_i^{(e)} \left(\frac{1}{Pe} \nabla^2 \phi^{(e)}(\mathbf{x},t) \right) \partial\Omega =& \frac{1}{Pe} \left(\int_{\Gamma^{(e)}} W_i^{(e)} (\nabla \phi^{(e)}(\mathbf{x},t).\vec{\mathbf{n}}) \ \partial\Gamma \right) \\ & - \frac{1}{Pe} \left(\int_{\Omega^{(e)}} \nabla W_i^{(e)} \cdot \nabla \phi^{(e)}(\mathbf{x},t) \ \partial\Omega \right) \end{aligned} \tag{5.3.24}$$

where $\vec{\mathbf{n}}$ is the normal direction at the boundary $\Gamma^{(e)}$ for this element. Substituting equation (5.3.24) into equation (5.3.23) yields

$$\begin{aligned} & \int_{\Omega^{(e)}} W_i^{(e)} \left(\frac{\partial \phi^{(e)}(\mathbf{x},t)}{\partial t} + \mathbf{v} \cdot \nabla \phi^{(e)}(\mathbf{x},t) \right) \partial\Omega + \frac{1}{Pe} \left(\int_{\Omega^{(e)}} \nabla W_i^{(e)} \cdot \nabla \phi^{(e)}(\mathbf{x},t) \partial\Omega \right) \\ & = \frac{1}{Pe} \left(\int_{\Gamma^{(e)}} W_i^{(e)} (\nabla \phi^{(e)}(\mathbf{x},t).\vec{\mathbf{n}}) \ \partial\Gamma \right) \end{aligned} \tag{5.3.25}$$

We next approximate the primary variable for this element $\phi^{(e)}(\mathbf{x},t)$ as follows

$$\phi^{(e)}(\mathbf{x},t) = \sum_{i=1}^{r} N_i^{(e)}(\mathbf{x}) \phi_i^{(e)}(t) \tag{5.3.26}$$

where $N_i^{(e)}(\mathbf{x})$ is the element shape function and $\phi_i^{(e)}(t)$ is the nodal solution of the element at time t. Substituting equation (5.3.26) into equation (5.3.25) yields the

following system of first order ordinary differential equations

$$\mathbf{M}^{(e)}\dot{\boldsymbol{\phi}}^{(e)} + (\mathbf{K_c}^{(e)} + \mathbf{K_d}^{(e)})\boldsymbol{\phi}^{(e)} = \mathbf{F}^{(e)} \tag{5.3.27}$$

where

$$\begin{aligned}
M_{ij}^{(e)} &= \int_{\Omega^{(e)}} W_i^{(e)} N_j^{(e)} \ \partial\Omega, \\
K_{c\,ij}^{(e)} &= \int_{\Omega^{(e)}} W_i^{(e)} \mathbf{v} \cdot \nabla N_j^{(e)} \ \partial\Omega, \\
K_{d\,ij}^{(e)} &= \frac{1}{Pe}\left(\int_{\Omega^{(e)}} \nabla W_i^{(e)} \cdot \nabla N_j^{(e)} \ \partial\Omega\right), \\
F_i^{(e)} &= \frac{1}{Pe}\left(\int_{\Gamma^{(e)}} W_i^{(e)} (\nabla \phi^{(e)}(\mathbf{x}, t).\vec{\mathbf{n}}) \ \partial\Gamma\right),
\end{aligned} \tag{5.3.28}$$

are the mass matrix, stiffness matrix due to convection, stiffness matrix due to diffusion, and force vector due to the Neumann boundary condition, respectively. Upon the assembly of all NE elements, we obtain the following global system of first order (in time) ordinary differential equations

$$\mathbf{M}\dot{\boldsymbol{\phi}} + (\mathbf{K_c} + \mathbf{K_d})\boldsymbol{\phi} = \mathbf{F} \tag{5.3.29}$$

where

$$\begin{aligned}
\mathbf{M} &= \sum_{e=1}^{NE} \mathbf{M}^{(e)}, & \mathbf{K_c} &= \sum_{e=1}^{NE} \mathbf{K_c}^{(e)} \\
\mathbf{K_d} &= \sum_{e=1}^{NE} \mathbf{K_d}^{(e)}, & \mathbf{F} &= \sum_{e=1}^{NE} \mathbf{F}^{(e)}
\end{aligned}$$

5.3.3 Time Integration by the GS4-1 Framework

In this section, we will provide more detailed validations on the robustness of the newly developed GS4-1 framework via the new selective control feature for applica-

tions of the convection-diffusion type. Using two numerical examples in one- and two-dimension governed by a linear transient convection-diffusion equation, we will show the importance of selectively controlling the numerical dissipation of the primary variable and its time derivative, which is a desirable attribute inherent to the GS4-1 framework in addition to the second order accuracy, zero-order overshoot, unconditional stability, and reduced computational expense due to solving the system of equations only once and requiring the solutions of only one previous time step.

5.3.3.1 1D Linear Convection-Diffusion

The first example is a one-dimensional convection-diffusion problem. The simplicity of the problem eliminates the need for upwinding in the spatial discretization procedure. The problem has the following initial and boundary conditions (in dimensionless form)

$$\phi(x,0) = \exp\left\{\frac{Pe}{2}(x-1)\right\} \tag{5.3.30}$$

$$\phi(0,t) = \exp\left\{-\frac{Pe}{2} - \frac{Pe}{4}t\right\} \tag{5.3.31}$$

$$\frac{\partial\phi}{\partial x}(1,t) = \frac{Pe}{2}\exp\left\{-\frac{Pe}{4}t\right\} \tag{5.3.32}$$

The analytical solutions in dimensionless form are given by [74]

$$\begin{aligned}\phi(x,t) &= \exp\left\{\frac{Pe(x-1)}{2} - \frac{Pe}{4}t\right\} \\ \dot{\phi}(x,t) &= -\frac{Pe}{4}\exp\left\{\frac{Pe(x-1)}{2} - \frac{Pe}{4}t\right\}\end{aligned} \tag{5.3.33}$$

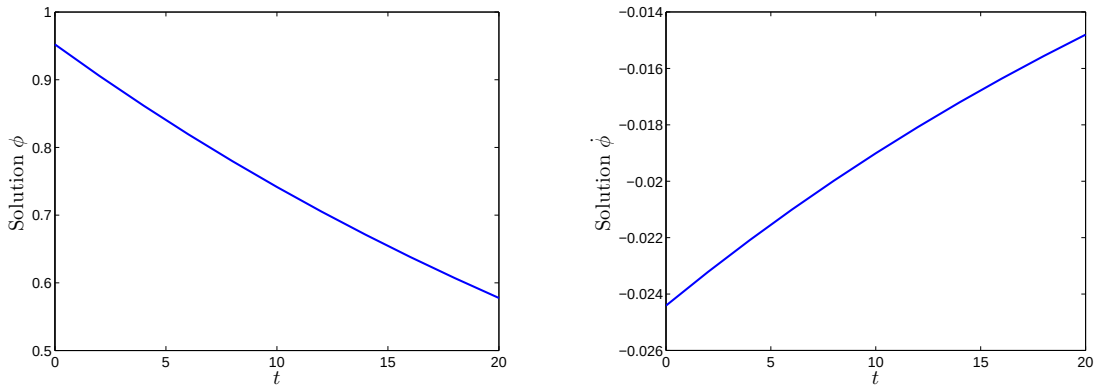
The problem parameters (in dimensionless form) are chosen as: $v = 1$ and $Pe = 0.1$. For this problem, we use 1D linear elements, whose element shape functions are given by

$$N_i^{(e)} = \begin{bmatrix} 1 - \frac{x}{l} & \frac{x}{l} \end{bmatrix} \tag{5.3.34}$$

where l is the length of each element. We discretize the spatial domain using

50 elements to yield a cell Peclet number of 0.002, in which case the Bubnov-Galerkin FEM can appropriately be used. In this method, the weighting function in equation (5.3.28) takes the same form as the element shape function (i.e., $W_i^{(e)} = N_i^{(e)}$) given by equation (5.3.34). We substitute this equation into equation (5.3.28) and assemble for all elements to form the global matrices of equation (5.3.29).

We solve the problem using the two cases under investigation with a time step size $\Delta t = 2$ and an end time $t = 20$. We first compare the performance of the two cases by looking at both the solutions of ϕ and $\dot{\phi}$ as a function of time for a specific node. For illustration purpose, we randomly choose a ρ_∞ value of 0.8 and a node at $x = 0.02$. Figure 5.24 shows the plots of analytical solutions (both ϕ and $\dot{\phi}$ as given by equation (5.3.33)) as a function of time for this node. Meanwhile, Figure 5.25 shows the plots of ϕ and $\dot{\phi}$ as a function of time for this node as generated by the GS4-1 framework with and without the selective control feature for the chosen ρ_∞ value. We can see from this figure that both cases yield good results for the primary variable (ϕ). However, it is clear from such a figure that the case without the selective control feature (i.e., algorithm with $\rho_\infty = \rho_\infty^s = 0.8$ which recovers the algorithm of the one-parameter method [62]) results in large oscillation in the solution of the time derivative variable ($\dot{\phi}$). The GS4-1 framework with the selective control feature, on the other hand, can suppress such oscillation and yields good agreement with the analytical solution for the same ρ_∞ value (see Figure 5.25(b)). This is possible due to its new selective control feature which allows the selection of different values for the parameters controlling the numerical dissipation of the two variables. By allowing the ρ_∞^s to take on a zero value via such a feature, regardless of the choice of ρ_∞ value, the GS4-1 framework guarantees that sufficient numerical damping is given to the time derivative variable such that the large oscillation is successfully eliminated and the algorithm yields physically representative solution as seen in Figure 5.25(b). This, however, cannot be achieved using the case with a single parameter (i.e., without the selective control feature) since this representation controls the numerical damping of both ϕ and $\dot{\phi}$ indiscriminately; hence, a ρ_∞ value of 0.8 means a corresponding ρ_∞^s value of 0.8 as well. This clearly demonstrates the advantage of GS4-1 framework with the new selective control feature in contrast to the one-parameter method (i.e., the case without the selective control feature).



(a) (b)

Figure 5.24: Plot of analytical solutions ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) as a function of time for a specific node at $x = 0.02$ as given by equation (5.3.33)

In addition to this solution history plots, we also look at the solutions of both ϕ and $\dot{\phi}$ for the whole spatial domain at a specific time. For this purpose, we again choose ρ_∞ value of 0.8 for consistency and a specific time of $t = 20$. Figure 5.26 shows the plots of analytical solutions of these variables (as given by equation (5.3.33)) as a function of the spatial domain at this specific time. Comparisons of the performance between the two cases in generating the ϕ and $\dot{\phi}$ solutions are shown in Figure 5.27. From this figure, we can see that the same observations as those seen in Figure 5.25 are repeated here, i.e., that (1) both cases yield good results for ϕ, (2) the one-parameter case (i.e., without the selective control feature) results in large oscillation in $\dot{\phi}$, and (3) the GS4-1 framework with the new selective control feature could yield good agreement with the analytical solution for the same ρ_∞ value. This, again, illustrates the importance of and the role played by the new feature introduced in the GS4-1 framework, which is the selective control of the numerical dissipation for ϕ and $\dot{\phi}$ not available in any existing methods for first order transient system to-date.

Since the previous observations seen in Figures 5.25 and 5.27 are based on the results generated using only one ρ_∞ value, we next compute and compare the errors in ϕ and $\dot{\phi}$ generated by these two cases for a set of ρ_∞ values ranging from 1 (nondissipative/zero

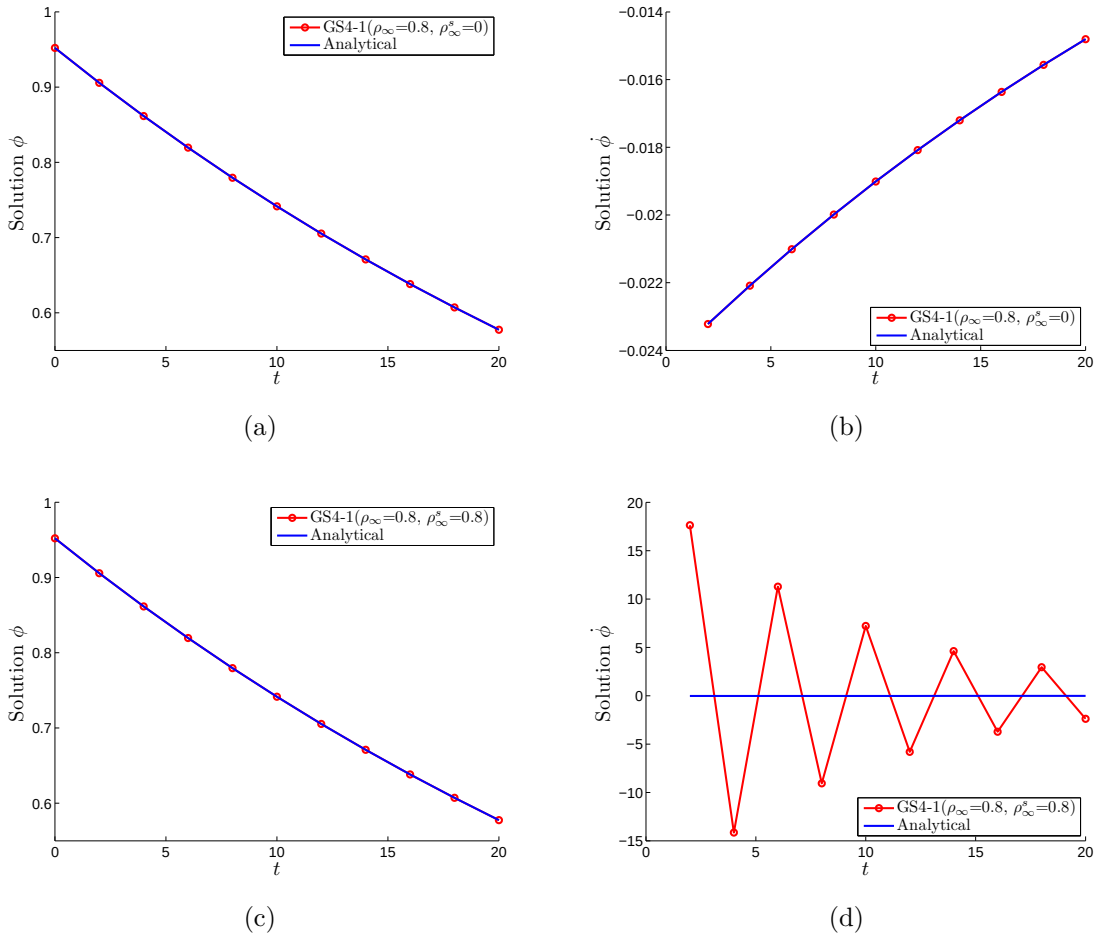
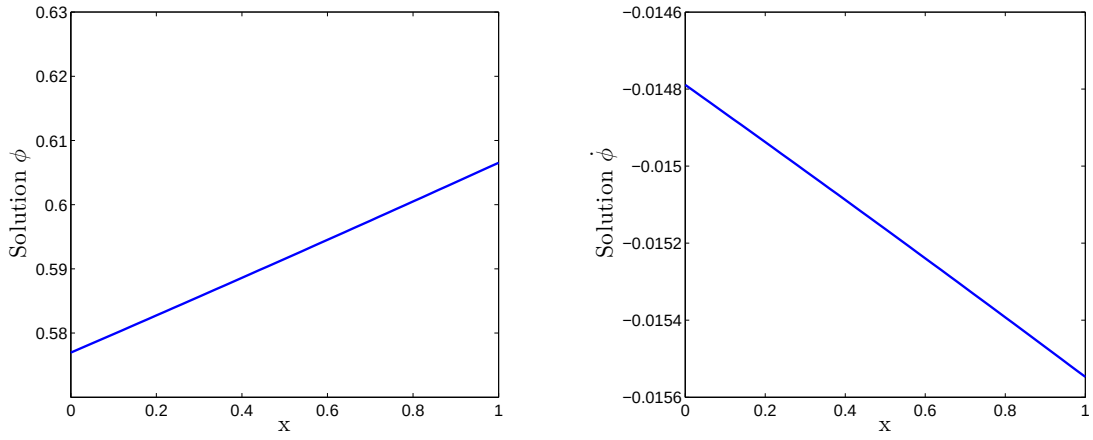


Figure 5.25: Plot of ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) as a function of time for a specific node at $x = 0.02$ generated by: (i) GS4-1($\rho_\infty = 0.8, \rho^s_\infty = 0$), and (ii) GS4-1($\rho_\infty = 0.8, \rho^s_\infty = 0.8$), i.e., the case without selective control features



(a) (b)

Figure 5.26: Plot of analytical solutions ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) as a function of x for a specific time of $t = 20$ as given by equation (5.3.33)

damping) to 0 (maximal numerical damping) in decrements of 0.1 for completion of the investigation. The error is defined as

$$Error = |Numerical - Analytical| \tag{5.3.35}$$

Table 5.3.3 first shows the comparison of maximal and total errors in the primary variable (ϕ) between the two cases for all ρ_∞ values considered. From this table, we can see that both cases perform well in generating numerical solution of ϕ. Meanwhile, Table 5.3.4 shows the comparison of the maximal and total errors in the time derivative variable ($\dot{\phi}$) between the two cases for all ρ_∞ values considered. In this table, the difference in performance between the two representations is obvious. The case without the selective control feature yields large errors (both maximal and total) for almost all ρ_∞ values considered except for large amount of numerical dissipation ($0 \leq \rho_\infty \leq 0.3$) which is less desirable. These large errors indicate that this method is not capable of eliminating the numerical oscillation associated with $\dot{\phi}$ for the given amount of numerical dissipation. This is due to the restriction of equal numerical dissipation for both ϕ and $\dot{\phi}$ imposed by this representation. It is apparent that the specified amount of the numerical dissipation is not sufficient to

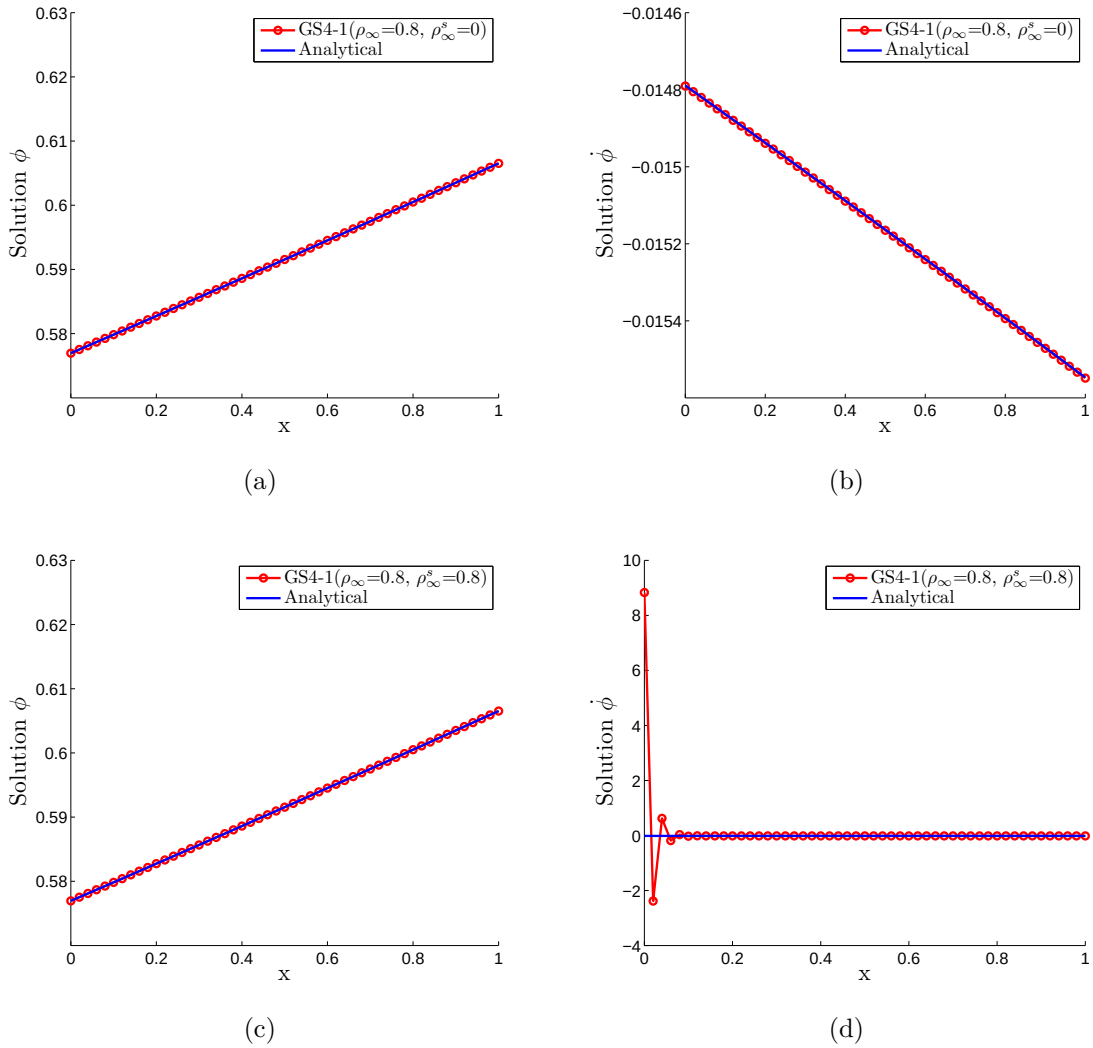


Figure 5.27: Plot of ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) as a function of x for a specific time of $t = 20$ generated by: (i) GS4-1($\rho_\infty = 0.8, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 0.8, \rho^s_\infty = 0.8$) i.e., the case without the selective control feature

suppress the numerical oscillation in $\dot{\phi}$ although the same amount of dissipation is enough to yield acceptable solution of ϕ. Such an observation indicates that different amounts of numerical dissipation are needed for these two variables. This requirement can easily be met by the two-parameter GS4-1 framework via the new selective control feature, which provides a more flexible and selective control of the numerical dissipation for ϕ and $\dot{\phi}$, respectively. As seen in Table 5.3.4, the case with the selective control feature successfully reduces the error in $\dot{\phi}$ to an acceptable level ($\times 10^{-6}$ for maximal error and $\times 10^{-5}$ for total error) for all ρ_∞ values considered.

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	3.9843x10^{-8}	3.9843x10^{-8}	1.1475x10^{-6}	1.1475x10^{-6}
0.9	4.2057x10^{-7}	3.8863x10^{-7}	1.4304x10^{-5}	1.3424x10^{-5}
0.8	5.2262x10^{-7}	3.5253x10^{-7}	1.8027x10^{-5}	1.2157x10^{-5}
0.7	3.6626x10^{-7}	2.9354x10^{-7}	1.2729x10^{-5}	1.0041x10^{-5}
0.6	2.3980x10^{-7}	2.8144x10^{-7}	8.3074x10^{-6}	9.5509x10^{-6}
0.5	2.0673x10^{-7}	3.0492x10^{-7}	7.0651x10^{-6}	1.0305x10^{-5}
0.4	2.2992x10^{-7}	3.4643x10^{-7}	7.7878x10^{-6}	1.1690x10^{-5}
0.3	2.7970x10^{-7}	3.9812x10^{-7}	9.4465x10^{-6}	1.3426x10^{-5}
0.2	3.5293x10^{-7}	4.5924x10^{-7}	1.1906x10^{-5}	1.5481x10^{-5}
0.1	4.6029x10^{-7}	5.3200x10^{-7}	1.5516x10^{-5}	1.7927x10^{-5}
0	6.2006x10^{-7}	6.2006x10^{-7}	2.0887x10^{-5}	2.0887x10^{-5}

Table 5.3.3: Comparison of error in ϕ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) the case without selective control with $\rho_\infty = \rho_\infty$, and (ii) the GS4-1 framework with selective control features with $\rho_\infty^s = 0$, for ρ_∞ values ranging from 1 (zero damping) to 0 (maximal damping) in decrements of 0.1.

These observations are reflected in Figures 5.28 through 5.32, which show the plots of errors in ϕ and $\dot{\phi}$ versus the spatial domain x generated by the two cases for all ρ_∞ values considered[3]. As seen in these figures, the large numerical oscillation in $\dot{\phi}$ generated by the case without the selective control feature can easily be turned off by the GS4-1 framework via the new selective control feature with the choice of $\rho_\infty^s = 0$. The ability to generate these physically representative solutions of both ϕ and $\dot{\phi}$ with small amount of numerical dissipation (ρ_∞ approaches unity) is a key

[3]except for $\rho_\infty = 0$ since the two cases yield the same algorithm, and therefore comparison of the errors is irrelevant

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	82.3789	$1.5406\text{x}10^{-6}$	112.5317	$7.6500\text{x}10^{-5}$
0.9	28.7237	$1.8570\text{x}10^{-6}$	39.2378	$8.8870\text{x}10^{-5}$
0.8	8.8454	$1.7989\text{x}10^{-6}$	12.0832	$8.6881\text{x}10^{-5}$
0.7	2.3270	$1.6990\text{x}10^{-6}$	3.1789	$8.3390\text{x}10^{-5}$
0.6	0.4981	$1.6409\text{x}10^{-6}$	0.6805	$8.1343\text{x}10^{-5}$
0.5	0.0805	$1.6181\text{x}10^{-6}$	0.1100	$8.0536\text{x}10^{-5}$
0.4	0.0086	$1.6115\text{x}10^{-6}$	0.0119	$8.0298\text{x}10^{-5}$
0.3	$4.8819\text{x}10^{-4}$	$1.6094\text{x}10^{-6}$	$7.4774\text{x}10^{-4}$	$8.0224\text{x}10^{-5}$
0.2	$9.4825\text{x}10^{-6}$	$1.6078\text{x}10^{-6}$	$6.4129\text{x}10^{-5}$	$8.0169\text{x}10^{-5}$
0.1	$2.0424\text{x}10^{-8}$	$1.6059\text{x}10^{-6}$	$8.3418\text{x}10^{-7}$	$8.0107\text{x}10^{-5}$
0	$1.6036\text{x}10^{-6}$	$1.6036\text{x}10^{-6}$	$8.0031\text{x}10^{-5}$	$8.0031\text{x}10^{-5}$

Table 5.3.4: Comparison of error in $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) the case without selective control with $\rho_\infty = \rho_\infty$, and (ii) the GS4-1 framework with selective control features with $\rho_\infty^s = 0$, for ρ_∞ values ranging from 1 (zero damping) to 0 (maximal damping) in decrements of 0.1.

desirable attribute of the GS4-1 framework via the new selective control feature. In this regard, we conclude that the GS4-1 framework with the selective control feature outperforms the existing method without such a feature.

5.3.3.2 2D Linear Convection-Diffusion

This problem serves to further provide a consistent observation for the two-dimensional case as well. The problem is defined in dimensionless form as follows:

$$\frac{\partial \phi}{\partial t} = \frac{1}{Pe}\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}\right) + v_x \frac{\partial \phi}{\partial x} + v_y \frac{\partial \phi}{\partial y} \tag{5.3.36}$$

on $\Omega = \{(x, y) | 0 \leq x, y \leq 1\}, t > 0$, with the following Dirichlet type boundary and initial conditions (in dimensionless form)

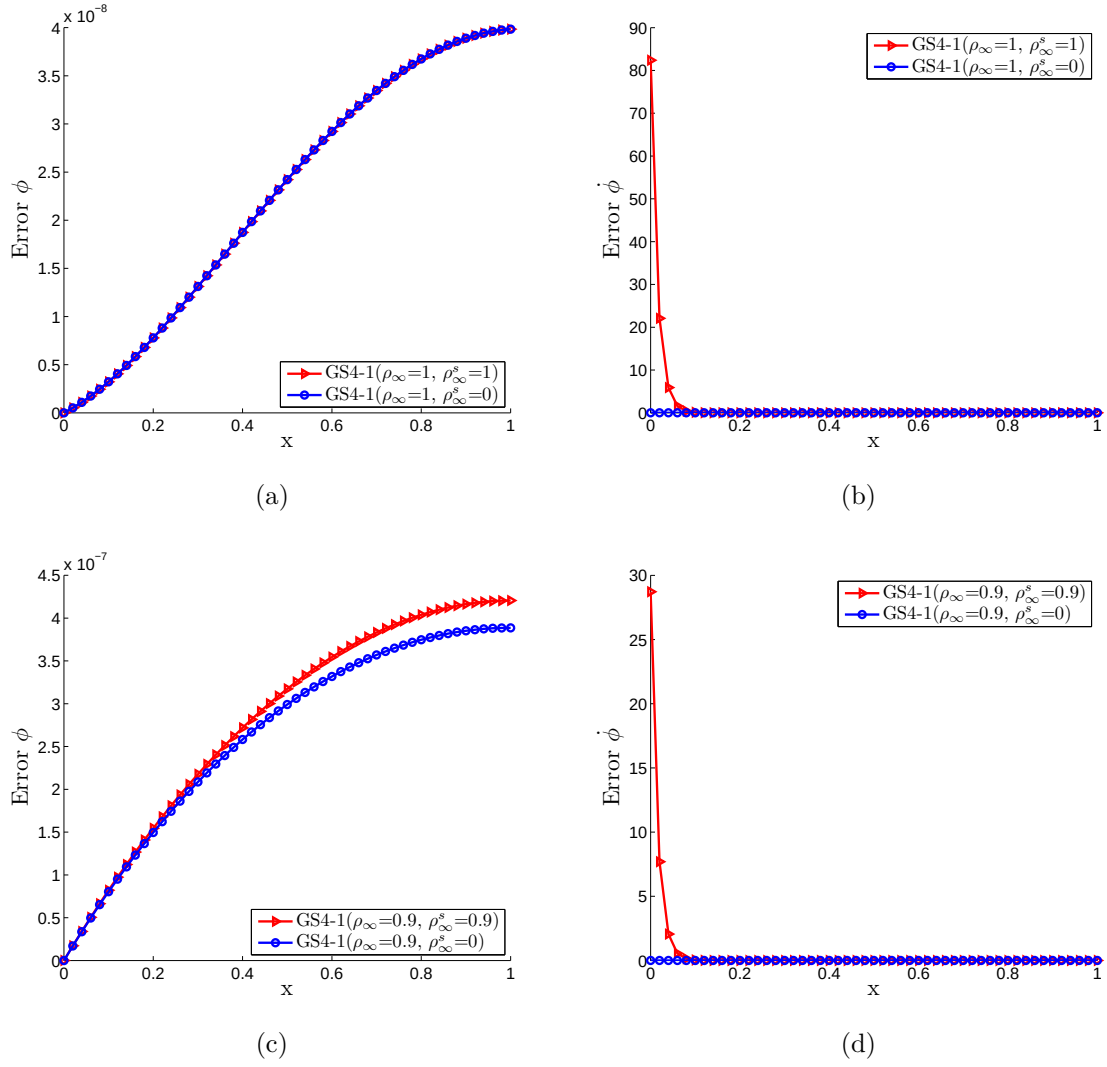


Figure 5.28: Comparison of error in ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$) i.e., the case without the selective control feature, and (ii) GS4-1(ρ_∞, $\rho^s_\infty = 0$), i.e., the case with the selective control feature, for ρ_∞ values of 1 and 0.9

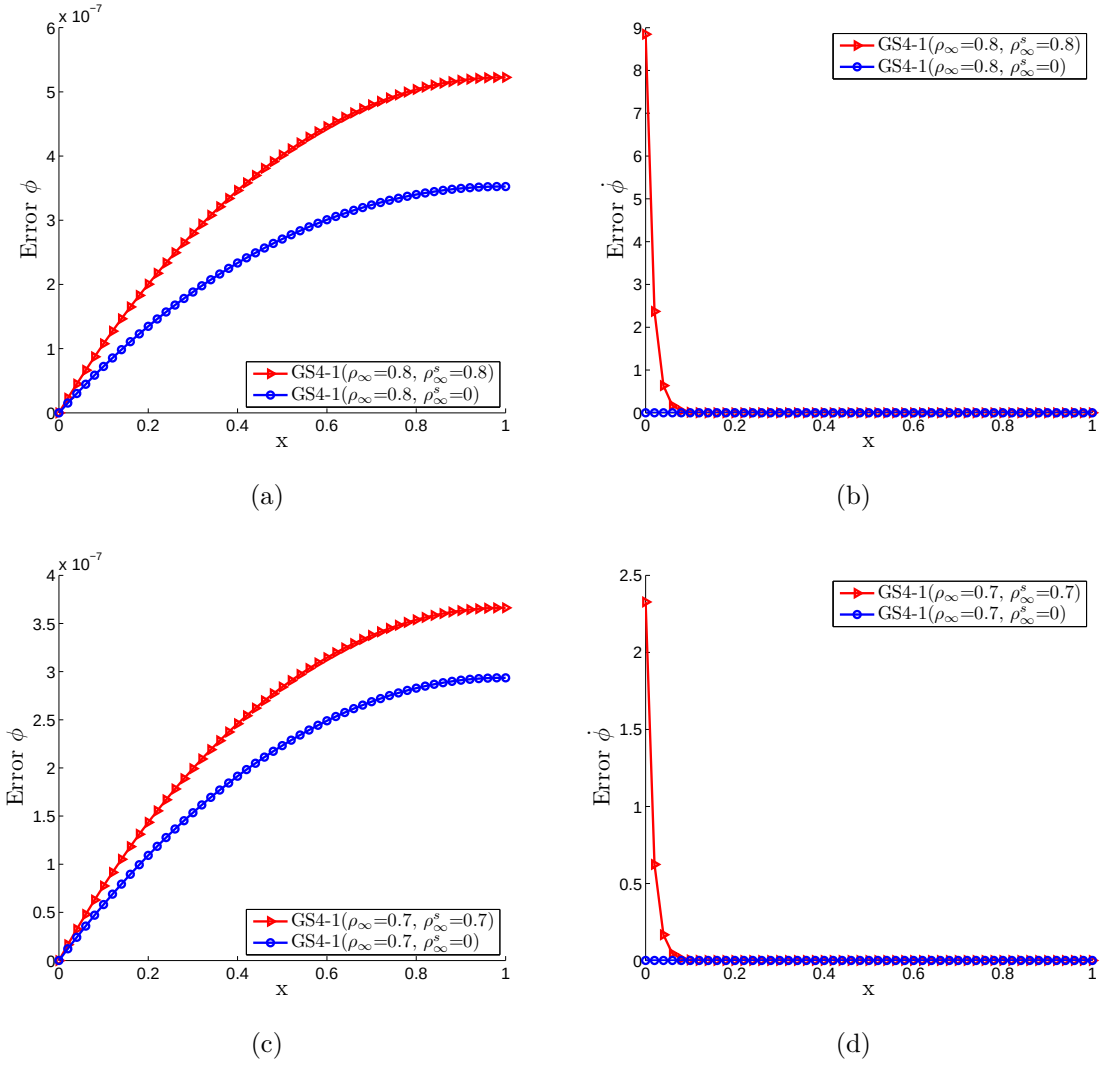


Figure 5.29: Comparison of error in ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$) i.e., the case without the selective control feature, and (ii) GS4-1(ρ_∞, $\rho^s_\infty = 0$), i.e., the case with the selective control feature, for ρ_∞ values of 0.8 and 0.7

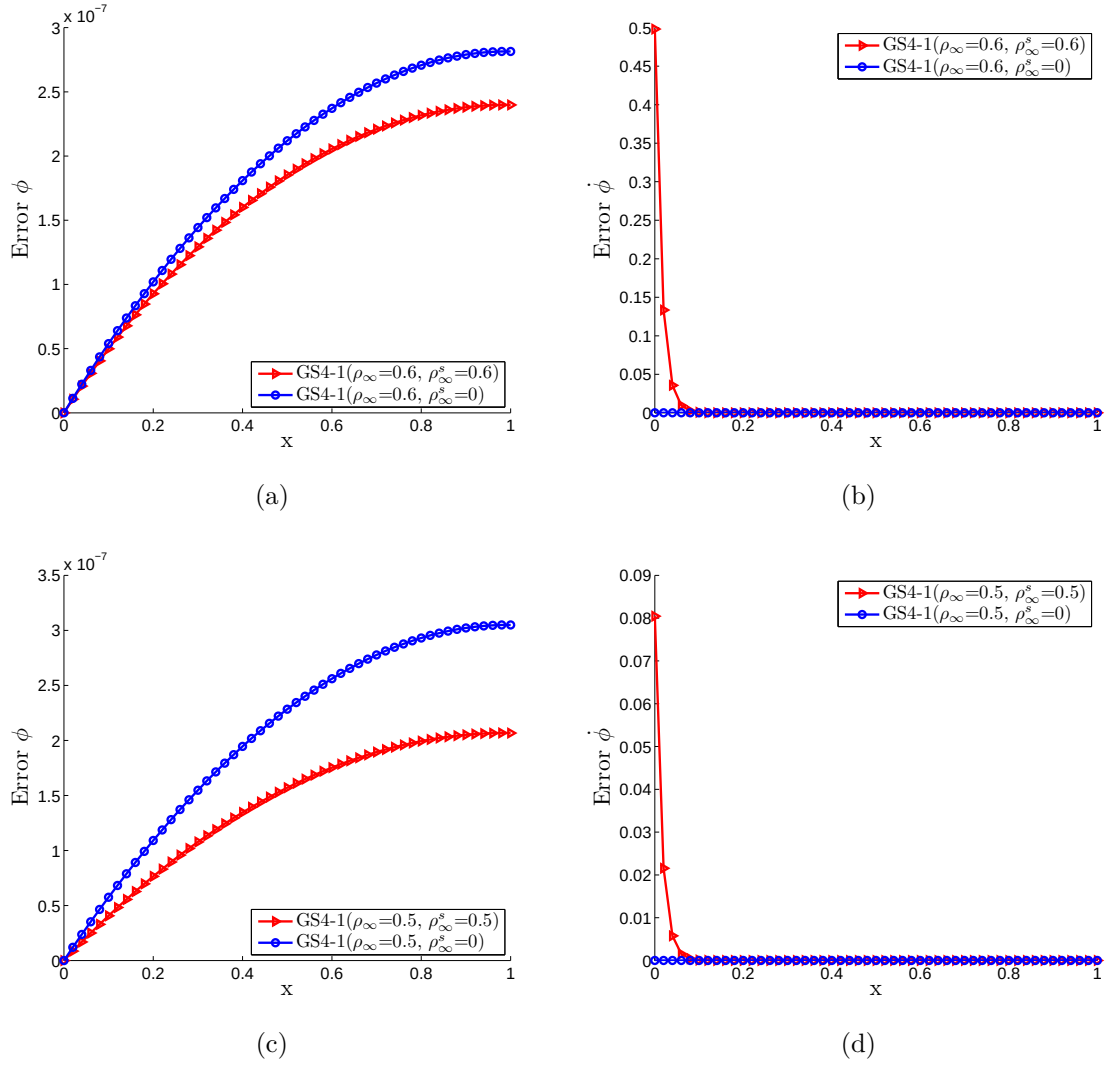


Figure 5.30: Comparison of error in ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$) i.e., the case without the selective control feature, and (ii) GS4-1(ρ_∞, $\rho^s_\infty = 0$), i.e., the case with the selective control feature, for ρ_∞ values of 0.6 and 0.5

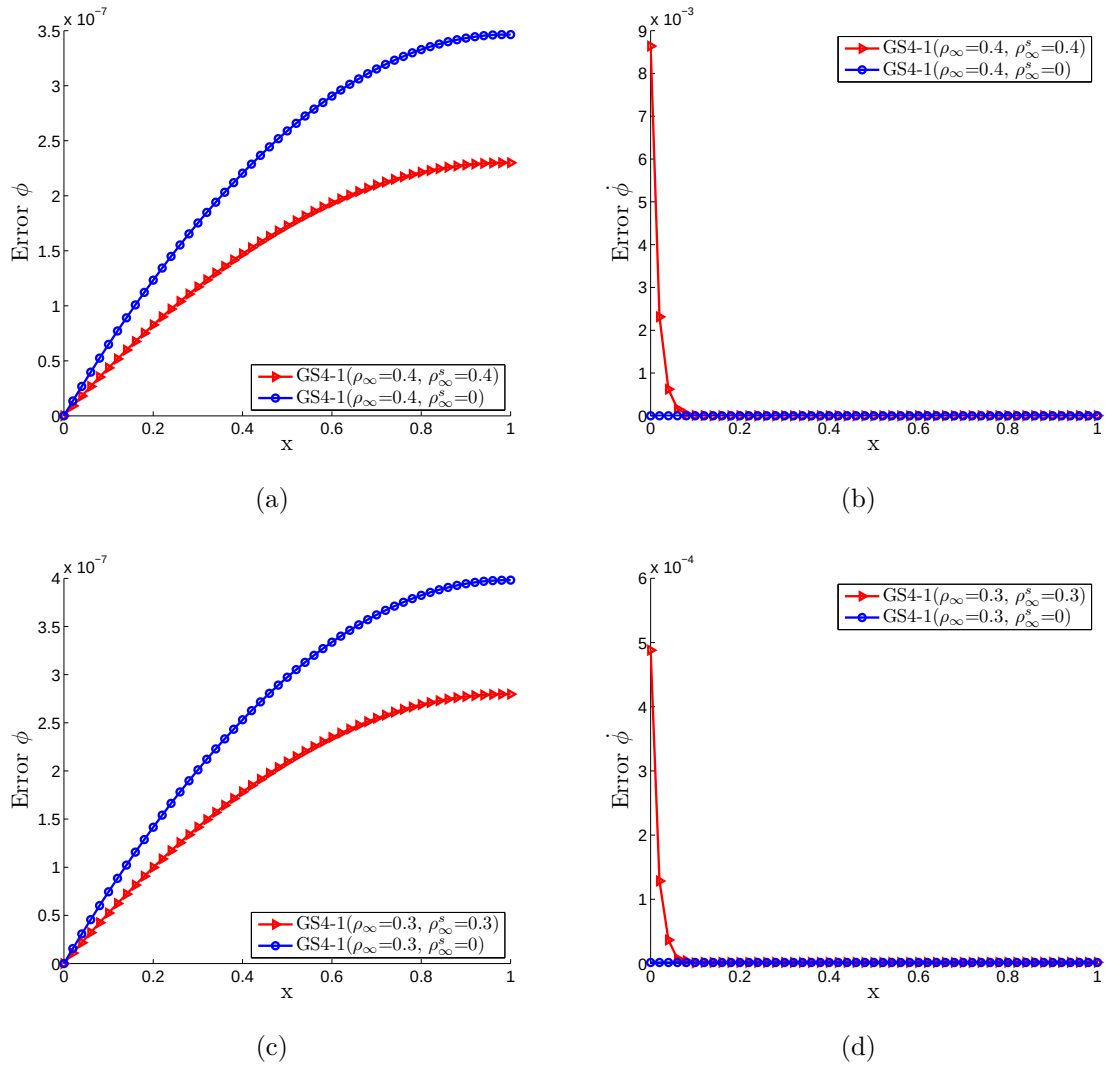


Figure 5.31: Comparison of error in ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$) i.e., the case without the selective control feature, and (ii) GS4-1(ρ_∞, $\rho^s_\infty = 0$), i.e., the case with the selective control feature, for ρ_∞ values of 0.4 and 0.3

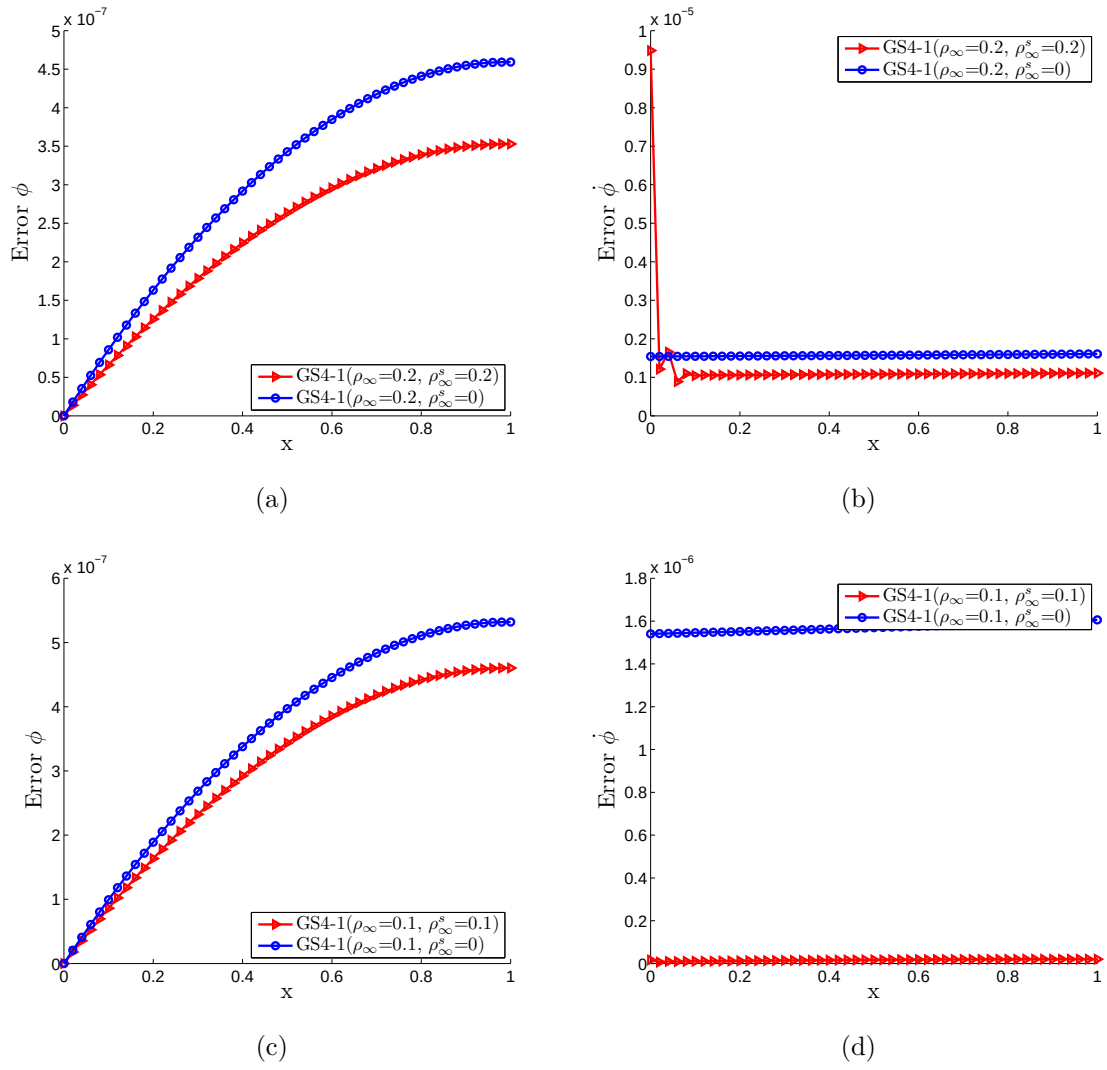


Figure 5.32: Comparison of error in ϕ and $\dot{\phi}$ (for the 1D convection-diffusion problem, Section 5.3.3.1) between: (i) GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$) i.e., the case without the selective control feature, and (ii) GS4-1(ρ_∞, $\rho^s_\infty = 0$), i.e., the case with the selective control feature, for ρ_∞ values of 0.2 and 0.1

$$
\begin{aligned}
\phi(0,y,t) &= \alpha e^{\beta t}(1+e^{-c_y y}) \\
\phi(x,0,t) &= \alpha e^{\beta t}(1+e^{-c_x x}) \\
\phi(1,y,t) &= \alpha e^{\beta t}(e^{-c_x}+e^{-c_y y}) \\
\phi(x,1,t) &= \alpha e^{\beta t}(e^{-c_x x}+e^{-c_y}) \\
\phi(x,y,0) &= \alpha(e^{-c_x x}+e^{-c_y y})
\end{aligned}
\tag{5.3.37}
$$

where the analytical solutions are given by [75] (in dimensionless form)

$$
\begin{aligned}
\phi(x,y,t) &= \alpha e^{\beta t}(e^{-c_x x}+e^{-c_y y}), \\
\dot{\phi}(x,y,t) &= \beta\alpha e^{\beta t}(e^{-c_x x}+e^{-c_y y}), \\
c_x &= \frac{Pe}{2}\left(v_x+\sqrt{v_x^2+4\beta/Pe}\right) \\
c_y &= \frac{Pe}{2}\left(v_y+\sqrt{v_y^2+4\beta/Pe}\right)
\end{aligned}
\tag{5.3.38}
$$

The constant physical properties and problem parameters (in dimensionless form) are chosen as: $v_x = v_y = 10$, $Pe = 10$, $\alpha = 1$, and $\beta = 0.1$. For this problem, we use 2D linear elements of width $2b$ and height $2h$ whose shape function is given by

$$
\begin{aligned}
N_i^{(e)} &= [N_1 \quad N_2 \quad N_3 \quad N_4] \\
N_1 &= \frac{(b-x)(h-y)}{4bh} \\
N_2 &= \frac{(b+x)(h-y)}{4bh} \\
N_3 &= \frac{(b+x)(h+y)}{4bh} \\
N_4 &= \frac{(b-x)(h+y)}{4bh}
\end{aligned}
\tag{5.3.39}
$$

We discretize the spatial domain using 20 x 20 elements. Because the resulting cell Peclet number is larger than 2, we use upwinding method in the spatial discretization. For this purpose, we employ the Streamline upwind/Petrov-Galerkin (SUPG) FEM

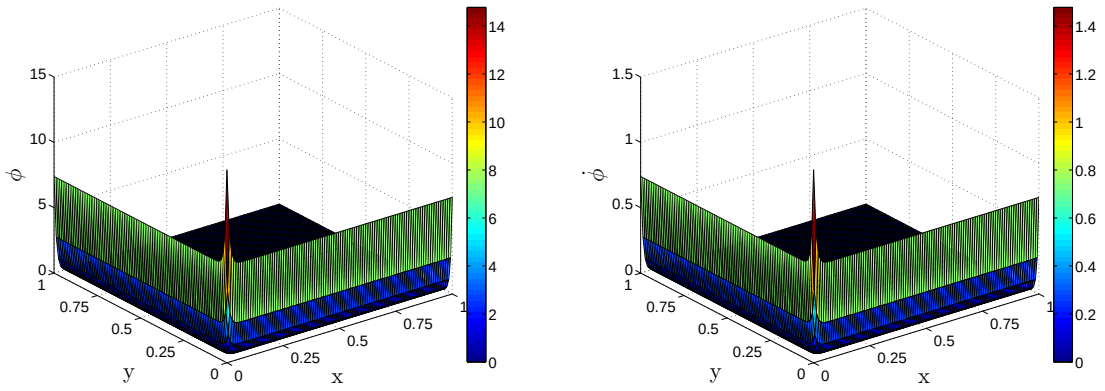
[76] in which the weighting function in equation (5.3.28) is given by

$$W_i^{(e)} = N_i^{(e)} + \frac{\hat{k}}{v_x^2 + v_y^2}\left(v_x \frac{\partial N_i^{(e)}}{\partial x} + v_y \frac{\partial N_i^{(e)}}{\partial y}\right) \tag{5.3.40}$$

where $\hat{k}$ is chosen to be [76]

$$\begin{aligned} \hat{k} =& \hat{\zeta} v_x b + \hat{\eta} v_y h \\ & \hat{\zeta} = (\coth \ \alpha_\zeta) - 1/\alpha_\zeta, \quad \hat{\eta} = (\coth \ \alpha_\eta) - 1/\alpha_\eta \\ & \alpha_\zeta = v_x b Pe, \quad \alpha_\eta = v_y h Pe \end{aligned} \tag{5.3.41}$$

We substitute equations (5.3.39) and (5.3.40) into equation (5.3.28) and assemble for all elements to form the global matrices of equation (5.3.29). We solve the problem using the two cases with a time step size of $\Delta t = 1$ and an end time of $t = 20$. The analytical solutions of ϕ and $\dot{\phi}$ for the entire spatial domain at this time are shown graphically in Figure 5.33.

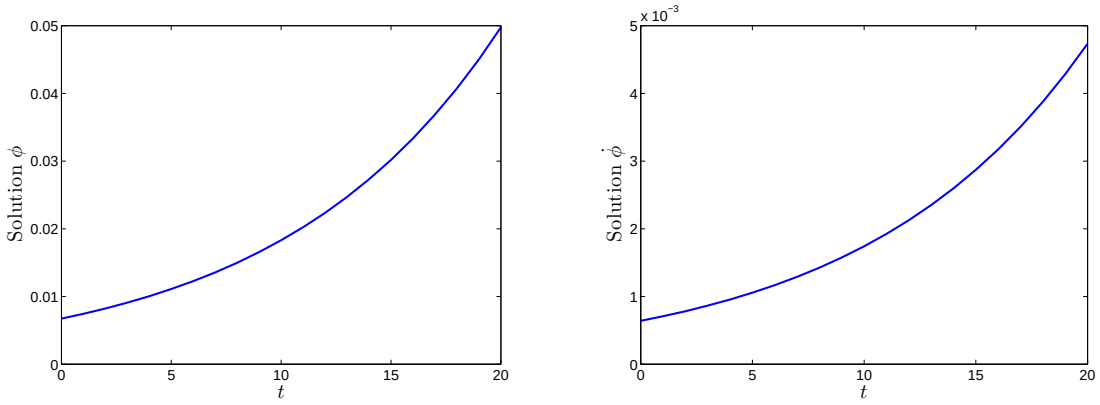


(a) (b)

Figure 5.33: 3D shaded surface plot of analytical solutions ϕ and $\dot{\phi}$ (for the 2D convection-diffusion problem, Section 5.3.3.2) at $t = 20$ as given by equation (5.3.38)

For this problem, we again compare the performance of these two cases by first

looking at the solutions of ϕ and $\dot{\phi}$ as a function of time for a specific node. For this purpose, we choose a ρ_∞ value of 0.8 and a node at $x = 0.05$ and $y = 0.15$. The analytical solutions, as given by equation (5.3.38), are shown in Figure 5.34. Meanwhile, the corresponding numerical solutions generated by the two cases are illustrated in Figure 5.35, which shows that both representations yield good results for ϕ. However, this figure also shows that the case without the selective control feature results in large oscillation in the time derivative variable ($\dot{\phi}$). This is due to the restriction in this representation which can only allow the same amount of numerical dissipation for both ϕ and $\dot{\phi}$, through the condition that ρ_∞^s is always equal to ρ_∞. Although the solution of ϕ may require only minimal numerical dissipation (ρ_∞ approaches 1), often, the solution of $\dot{\phi}$ requires more numerical dissipation. Therefore, although this representation could yield good result for ϕ, often, it may not be able to eliminate the numerical oscillation associated with $\dot{\phi}$ given the minimal numerical dissipation. The GS4-1 framework with the selective control feature, on the other hand, yields good agreement with the analytical solutions for both ϕ and $\dot{\phi}$ with the same ρ_∞ value. It is achieved via its new selective control feature as previously explained. This is clearly the advantage of the GS4-1 framework involving such a new feature over the existing method without such a feature.



(a) (b)

Figure 5.34: Plot of analytical solutions ϕ and $\dot{\phi}$ (for the 2D convection-diffusion problem, Section 5.3.3.2) as a function of time for a specific node at $x = 0.05$ and $y = 0.15$ as given by equation (5.3.38)

We repeat the same procedure to determine the errors in ϕ and $\dot{\phi}$ as previously done

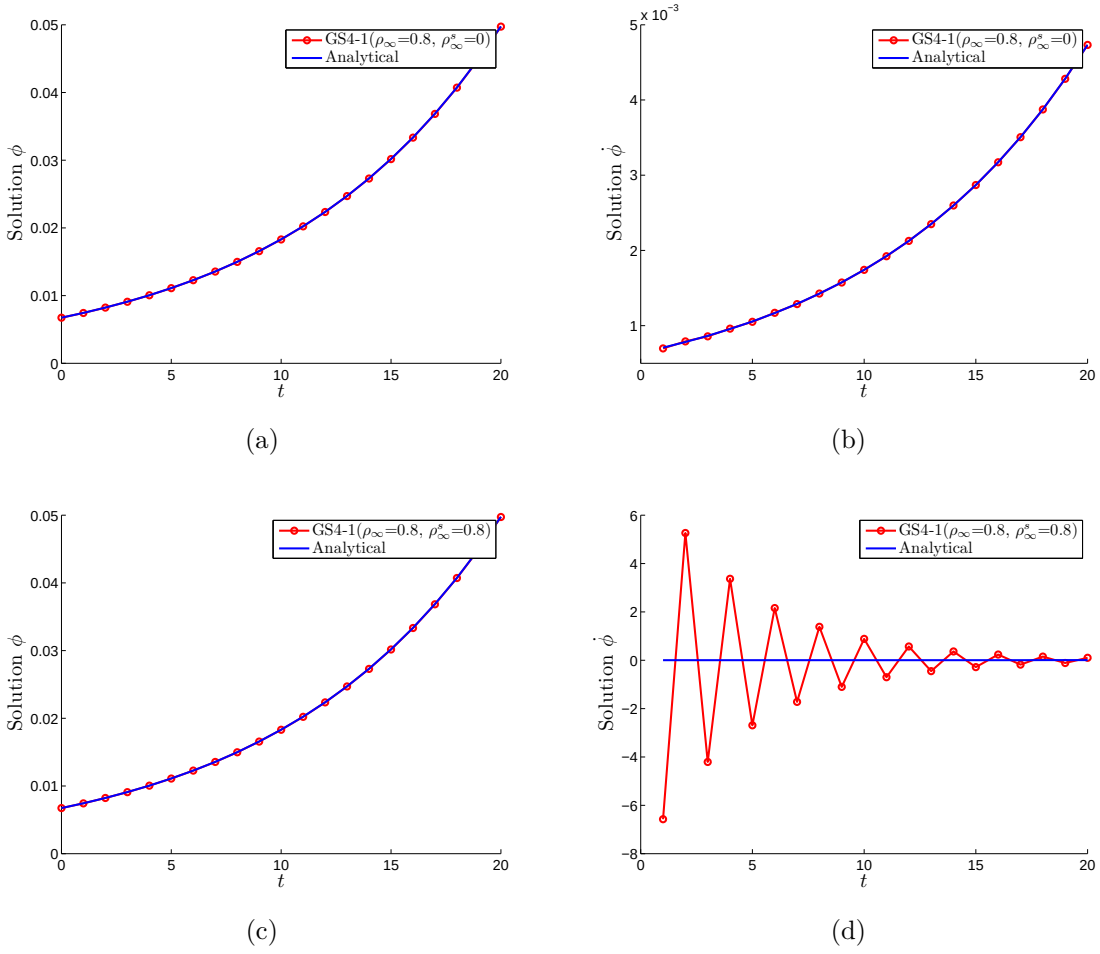


Figure 5.35: Plot of ϕ and $\dot{\phi}$ (for the 2D convection-diffusion problem, Section 5.3.3.2) as a function of time for a specific node at $x = 0.05$ and $y = 0.15$ generated by: (i) GS4-1($\rho_\infty = 0.8, \rho^s_\infty = 0$), i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 0.8, \rho^s_\infty = 0.8$), i.e., the case without the selective control feature

in the 1D case (Section 5.3.3.1) and show the results in Table 5.3.5 and 5.3.6. An observation of Table 5.3.5 indicates that the errors in ϕ generated by the two cases are of the same order of magnitude and are small (10^{-5} for maximal error and 10^{-3} for total error). This supports our previous observations that both cases (i.e., the GS4-1 framework with and without the selective control feature) work well to yield physically representative solutions of the primary variable (ϕ). On the other hand, the errors in $\dot{\phi}$ (both maximal and total) generated by the case without the selective control feature are large as seen in Table 5.3.6, except for large amount of numerical dissipation ($0 \leq \rho_\infty \leq 0.5$) which is less desirable (due to possible over-dissipation as discussed before). These errors are significantly reduced to an acceptable level (10^{-4} for maximal error and 10^{-2} for total error) by the case with the selective control feature. It is hence apparent that the previous observations seen in the 1D case are also valid in this two-dimensional case. This provides validation on the consistency of the argument. Therefore, we again conclude that the GS4-1 framework with the new selective control feature is better than the existing methods in the same class (without such a feature).

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	7.7101x10^{-5}	7.7101x10^{-5}	0.0014	0.0014
0.9	8.2838x10^{-5}	8.4817x10^{-5}	0.0015	0.0015
0.8	8.5482x10^{-5}	8.5802x10^{-5}	0.0015	0.0015
0.7	8.5896x10^{-5}	8.5883x10^{-5}	0.0015	0.0015
0.6	8.5925x10^{-5}	8.5866x10^{-5}	0.0015	0.0015
0.5	8.5911x10^{-5}	8.5840x10^{-5}	0.0015	0.0015
0.4	8.5887x10^{-5}	8.5810x10^{-5}	0.0015	0.0015
0.3	8.5853x10^{-5}	8.5775x10^{-5}	0.0015	0.0015
0.2	8.5804x10^{-5}	8.5736x10^{-5}	0.0015	0.0015
0.1	8.5735x10^{-5}	8.5689x10^{-5}	0.0015	0.0015
0	8.5635x10^{-5}	8.5635x10^{-5}	0.0015	0.0015

Table 5.3.5: Comparison of error in ϕ (for the 2D convection-diffusion problem, Section 5.3.3.2) between: (i) the case without the selective control feature ($\rho_\infty^s = \rho_\infty$), and (ii) GS4-1 framework with the selective control feature (with $\rho_\infty^s = 0$), for ρ_∞ values ranging from 1 (zero damping) to 0 (maximal damping) in decrements of 0.1.

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	843.7890	$5.8580\text{x}10^{-4}$	$1.7808\text{x}10^{4}$	0.0126
0.9	102.5860	$5.8580\text{x}10^{-4}$	$2.1651\text{x}10^{3}$	0.0123
0.8	9.7294	$5.8580\text{x}10^{-4}$	205.3404	0.0124
0.7	0.6744	$5.8580\text{x}10^{-4}$	14.2336	0.0124
0.6	0.0319	$5.8580\text{x}10^{-4}$	0.6737	0.0124
0.5	0.0018	$5.8580\text{x}10^{-4}$	0.0375	0.0124
0.4	$8.6103\text{x}10^{-4}$	$5.8580\text{x}10^{-4}$	0.0183	0.0124
0.3	$6.5438\text{x}10^{-4}$	$5.8580\text{x}10^{-4}$	0.0140	0.0124
0.2	$3.7513\text{x}10^{-4}$	$5.8580\text{x}10^{-4}$	0.0081	0.0124
0.1	$2.0577\text{x}10^{-5}$	$5.8580\text{x}10^{-4}$	$5.7624\text{x}10^{-4}$	0.0124
0	$5.8580\text{x}10^{-4}$	$5.8580\text{x}10^{-4}$	0.0124	0.0124

Table 5.3.6: Comparison of error in $\dot{\phi}$ (for the 2D convection-diffusion problem, Section 5.3.3.2) between: (i) the case without the selective control feature ($\rho^s_\infty = \rho_\infty$), and (ii) GS4-1 framework with the selective control feature (with $\rho^s_\infty = 0$), for ρ_∞ values ranging from 1 (zero damping) to 0 (maximal damping) in decrements of 0.1.

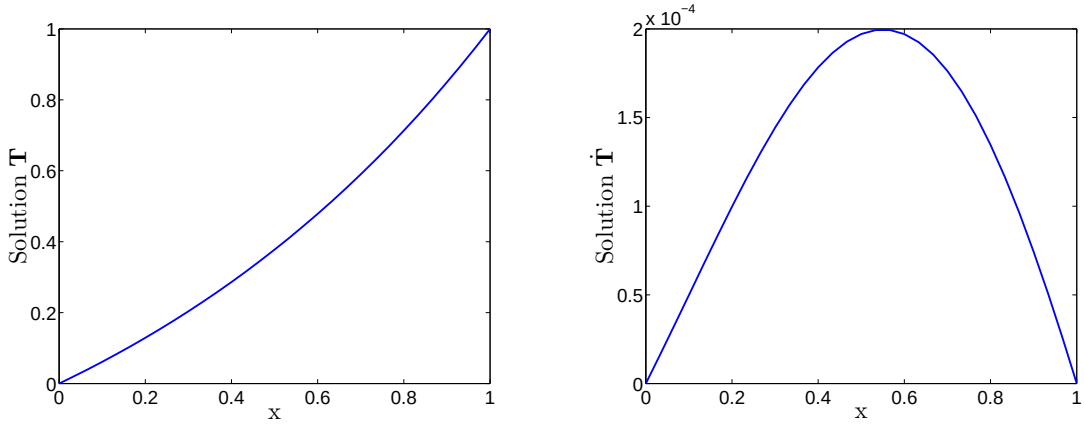
5.4 Convective Heat Transfer with Thermal Shock

Convective heat transfer is one of the major modes of heat transfer, other than conduction and radiation. It is a mechanism of heat transfer occurring because of bulk fluid motion. The heat energy being transferred between the moving fluid and a surface at different temperatures is known as convection. Such a mechanism is governed by the convection-diffusion equation described previously in Section 5.3 (hence, the governing equation for this problem as well as the spatial discretization procedure will not be repeated here). In particular, for the convective heat transfer, heat (temperature) is the entity of interest being advected (carried), and diffused (dispersed). Heat transfer by convection occurs in numerous examples of naturally occurring fluid flow, such as: wind, oceanic currents, and movements within the Earth's mantle. In engineering practices, it is used to provide desired temperature changes, as in heating of homes, industrial processes, cooling of equipment, etc. Due to its importance in various natural and engineering applications, we consider in this section an application of the newly developed GS4-1 framework to a particularly challenging problem arising in these contexts.

In particular, we consider an example involving a thermal shock due to a sudden change in the temperature at the boundary. For such a case, the primary variable (temperature) would typically exhibit numerical oscillations due to the presence of the thermal shock. Note that this is in contrast to the previous examples in Section 5.3 which do not involve thermal shock and therefore the numerical solutions of the primary variable in these cases agree well with the analytical solution even without or with little numerical dissipation (see Figures 5.25, 5.27, and 5.35). Consequently, for the problems involving thermal shock, typically the nondissipative scheme fails and numerically dissipative schemes are employed to stabilize the numerical solution of the primary variable. With such a requirement, this problem serves as a good numerical example to fully illustrate the ability and advantage of the newly developed GS4-1 framework and in particular the role played by the new selective control feature.

5.4.1 Time Discretization by the GS4-1 Framework

As done previously, we solve the problem using the two cases, i.e., (i) GS4-1 framework with the selective control feature, and (ii) GS4-1 framework without the selective control feature, as defined in *Remark* 4.3.1. For each case, we solve the one-dimensional problem of convective heat transfer (in dimensionless form, where the governing equation is that described in Section 5.3, i.e., equation (5.3.22)) using a time step size of $\Delta t = 0.1$ and an end time of $t = 1$. The problem parameters (in dimensionless form) are: $v = 1$ and $Pe = 1$, with an initial temperature of $T_0 = 0$. To impose a thermal shock to the problem, we enforce a sudden change in the temperature at the right boundary ($x = 1$) through a Dirichlet boundary condition $T(x = 1, t > 0) = 1$. The spatial domain is discretized using the Galerkin Finite Element Method with a total of 30 elements. Figure 5.36 shows the corresponding analytical solutions of the primary variable (T) and the time derivative variable ($\dot{T}$), as given in [77].



(a) (b)

Figure 5.36: Analytical solution of T and $\dot{T}$ (for the convective heat transfer problem with thermal shock, Section 5.4) as a function of x at $t = 1$

As previously described in Section 4.4, in solving any transient problem of interest, one would first use the nondissipative algorithm since numerical dissipation should be imposed only if such nondissipative algorithm fails or, in other words, results in highly oscillatory solutions. With this in mind, we first solve the problem using $\rho_\infty = 1$. This means that the case without the selective control feature is defined

as: GS4-1($\rho_\infty = \rho^s_\infty = 1$), in which case the framework recovers the existing nondissipative algorithm for first order transient systems, i.e., the Crank-Nicolson method. Meanwhile, the other case with the selective control feature is defined as: GS4-1($\rho_\infty = 1$,$\rho^s_\infty = 0$). The former definition of the algorithm indicates that no numerical dissipation is imposed on both the primary variable and its time derivative, i.e., the resulting algorithm is totally nondissipative (the Crank-Nicolson method). Meanwhile, the latter definition of the algorithm means that no numerical dissipation is imposed on the primary variable but the time derivative variable is intentionally given maximal numerical dissipation due to the reasons cited previously.

We first compare the performance of the two cases by looking at the solutions of T and $\dot{T}$ generated by these two cases for the whole spatial domain (i.e., as a function of x) for a specific time of $t = 1$. The numerical solutions generated by the two cases defined above are illustrated in Figure 5.37. As expected, the numerical solution of the primary variable (T) generated by the nondissipative algorithm (i.e., $\rho_\infty = 1$)[4] is oscillating (see Figure 5.37(a)), due to the sudden change in the temperature at the right boundary (i.e., thermal shock). In this situation, therefore, numerical dissipation for this variable is needed to obtain physically representative solution. An observation of Figure 5.37(b) indicates that imposing no numerical dissipation on the time derivative variable ($\dot{T}$) by the case without the selective control feature (i.e., by selecting $\rho^s_\infty = 1$) also results in numerical instabilities in this variable. On the other hand, the case with the selective control feature is able to eliminate such numerical instabilities as seen in Figure 5.37(b)[5]. It is worthy to note here that although the physically representative solution of $\dot{T}$ can be obtained using such a representation, the generated numerical solution of T is still not satisfying with the current choice of $\rho_\infty = 1$.

Therefore, we now proceed to solve the problem using dissipative algorithms,

[4]note from Figure 5.37(a) that both cases yield the same solution of T since $\rho_\infty = 1$, in which case the ρ^s_∞ value does not have any effect at all on the numerical solution of T as discussed before (see Section 4.1.3), hence demonstrates the consistency.

[5]it is to be noted that due to the automatic scale generated by the plotting routine employed, although not representative, this oscillation makes the numerical solutions generated by the case with the selective control feature simply appear as a straight line due to the different ranges on the $\dot{T}$ solution values

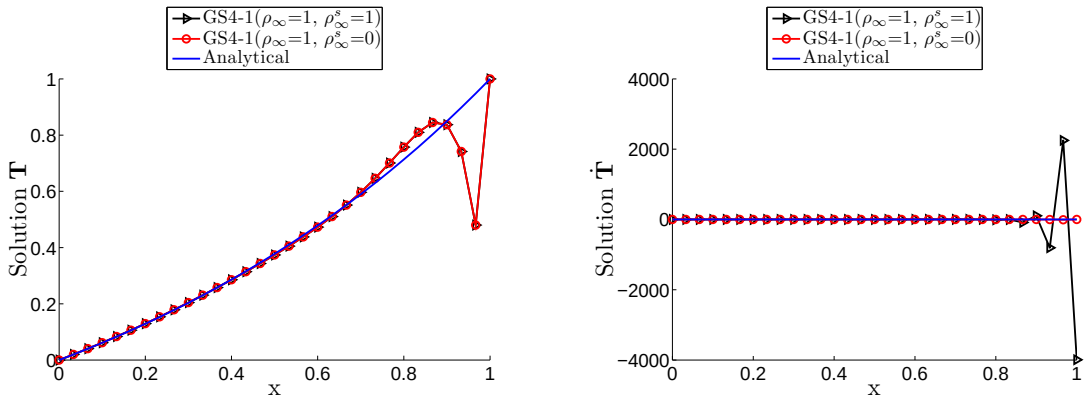
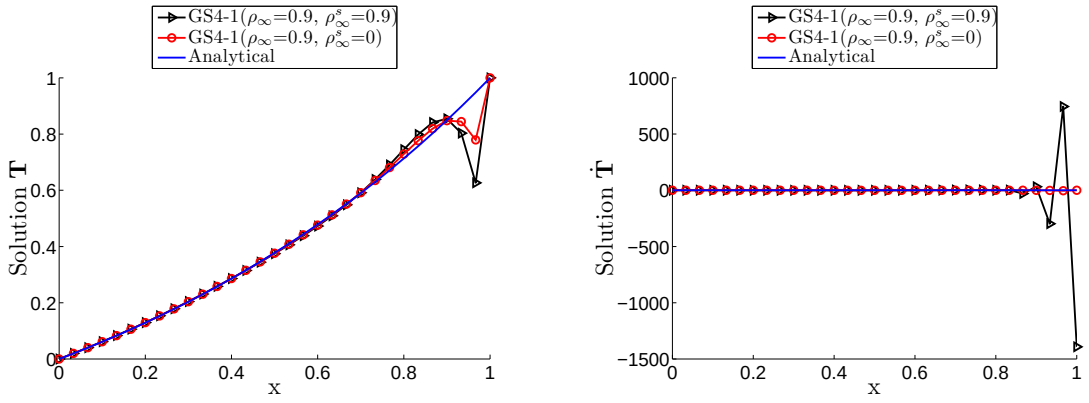


Figure 5.37: Plot of T and $\dot{T}$ (for the convective heat transfer problem with thermal shock, Section 5.4) as a function of x at $t = 1$ using $\Delta t = 0.1$, generated by: (i) GS4-1($\rho_\infty = 1, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = \rho^s_\infty = 1$) i.e., the case without the selective control feature.

which are available within the GS4-1 framework by choosing ρ_∞ value of less than unity (i.e., $0 \leq \rho_\infty < 1$).[6] As previously highlighted, this however must be done in a strategic manner as outlined in Section 4.4 since we do not wish to impose over-dissipation that may deviate the numerical solutions from the actual dynamics of the problem. For this purpose, we consider solving the problem using the two cases with $\rho_\infty = 0.9$. This choice of ρ_∞ means that a slight numerical dissipation is imposed on the primary variable. For the case without the selective control feature, an equal amount of numerical dissipation is imposed on the time derivative variable, which indicates the disadvantage of the method since Figure 5.37(b) clearly indicates that this variable needs greater amount of numerical dissipation to eliminate the oscillation. The results are shown in Figure 5.38. By introducing numerical dissipation to the primary variable, we can see from Figure 5.38(a) that the oscillation in this variable is reduced as compared to the case when no numerical dissipation is imposed (see Figure 5.37(a)). More interestingly, it is to be noted from Figure 5.38(a) that the oscillation in T generated by the case with the selective control feature is less than that generated by the case without such a feature. This demonstrates that the new selective control feature plays a role towards obtaining

[6]Recall that a value of 1 means nondissipative/zero damping, while a value of zero means maximal numerical damping.

accurate solutions of, not only the time derivative variable, but also the primary variable. Meanwhile, the solution of $\dot{T}$ generated by the case without the selective control feature is still greatly oscillating as seen in Figure 5.38(b). This is as expected since this method only imposes slight numerical dissipation on this variable with a choice of $\rho_\infty = \rho_\infty^s = 0.9$ whilst it is clear from Figure 5.37(b) that this variable needs greater amount of numerical dissipation.



(a) (b)

Figure 5.38: Plot of T and $\dot{T}$ (for the convective heat transfer problem with thermal shock, Section 5.4) as a function of x at $t = 1$ using $\Delta t = 0.1$, generated by: (i) GS4-1($\rho_\infty 0.9, \rho_\infty^s = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = \rho_\infty^s = 0.9$) i.e., the case without the selective control feature

As the numerical oscillation in T is still present with the current choice of ρ_∞ value, we further increase the numerical dissipation on this variable by reducing ρ_∞ value until a physically representative solution is obtained. Figure 5.39 shows the results for T and $\dot{T}$ generated by the two cases, i.e., the GS4-1 framework with and without the selective control feature, for ρ_∞ values of 0.8 and 0.7. Similar trends of performance of the two cases, as described previously for the case of $\rho_\infty = 0.9$ (Figure 5.38), can be seen from this figure, however with the oscillations in both T and $\dot{T}$ being reduced (as expected) as more numerical dissipation is imposed. From our investigations, the physically representative solutions of the primary variable T can be obtained with a minimal numerical dissipation of $\rho_\infty = 0.6$ as depicted in Figure 5.40(a). It is to be noted, however, that the oscillation in $\dot{T}$ is still present

for the case without the selective control feature as seen in Figure 5.40(b). The present development with the selective control feature, on the other hand, is able to yield physically representative solutions of, not only the primary variable, but also the time derivative one from the choice of $\rho_\infty^s = 0$. The results shown and discussed above demonstrate the significant importances and effects of the selective control feature in both the primary variable and its time derivative, towards obtaining physically representative solutions, in contrast to the other existing methods.

For completeness of the analysis, we next explicitly demonstrate the rate of convergence in time (i.e., the order of accuracy) of the algorithms in the GS4-1 framework for the example presented in this section. For this purpose and for consistency, we show the convergence plots of some of the algorithms used in obtaining the numerical results shown above, namely, GS4-1 framework with: (i) $\rho_\infty = 1$, $\rho_\infty^s = 0$, (ii) $\rho_\infty = 0.9$, $\rho_\infty^s = 0$, (iii) $\rho_\infty = 0.7$, $\rho_\infty^s = 0$, and (iv) $\rho_\infty = 0.6$, $\rho_\infty^s = 0$. The total number of time steps (N_i, see equation (4.5.45)) used to construct the plots are chosen as 10000, 500, 200, and 50, where the solutions generated with $N_i = 10000$ are used as the reference to calculate the errors for the construction of the convergence plots. Meanwhile, the end time at which the solutions are evaluated is $t_{end} = 1$. Figure 5.41 shows the convergence plots for T and $\dot{T}$ generated by these GS4-1 algorithms. In contructing these convergence plots, we use the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$, which is the proper construction of the convergence plot as discussed in Section 4.5. We can see from the convergence plots in this figure that these algorithms are all second order accurate; hence demonstrate such a desirable attribute of the developed GS4-1 framework.

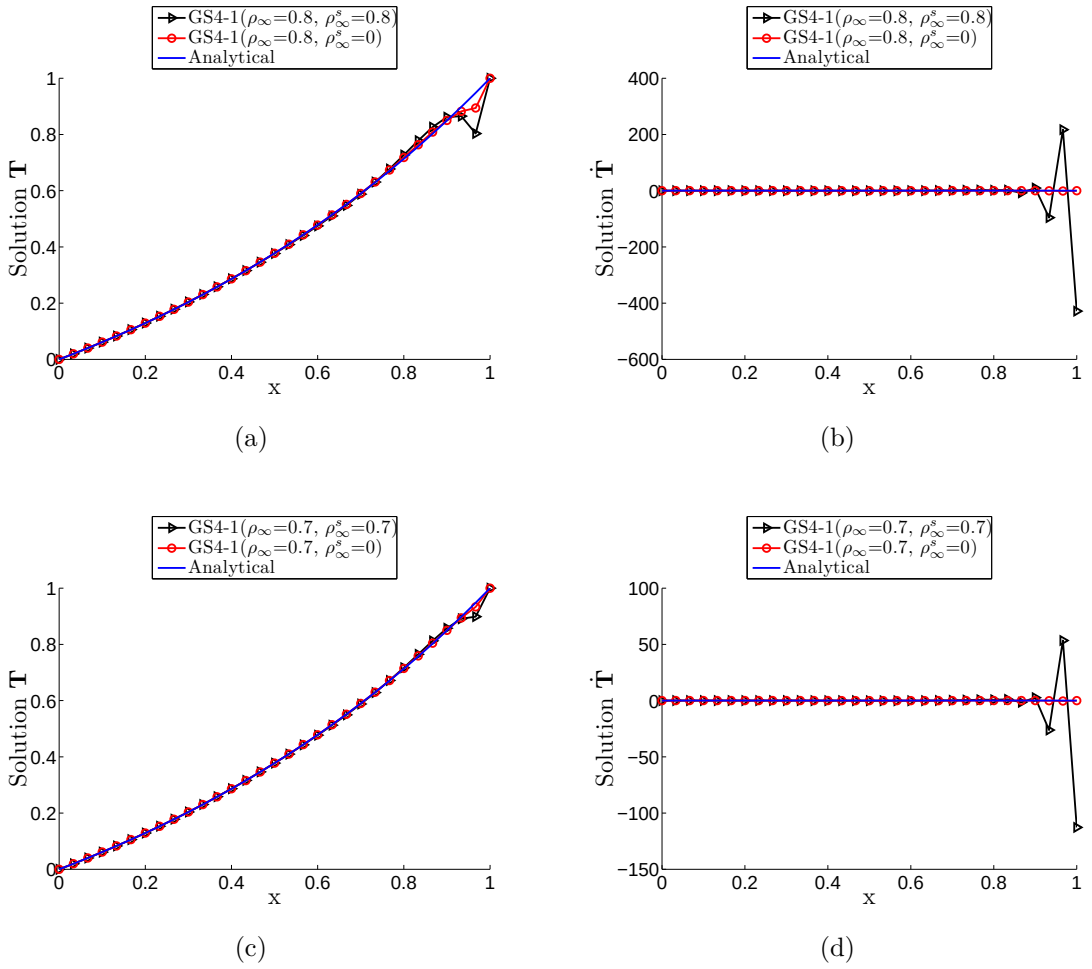
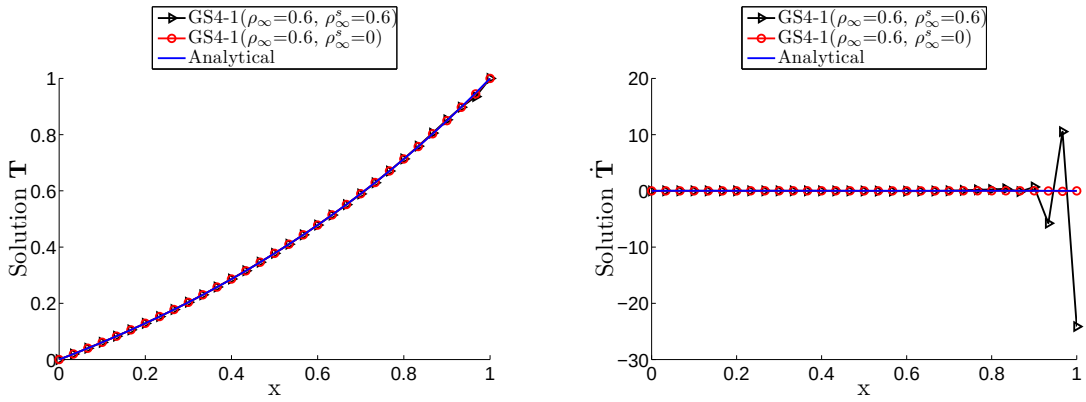


Figure 5.39: Plot of T and $\dot{T}$ (for the convective heat transfer problem with thermal shock, Section 5.4) as a function of x at $t = 1$ using $\Delta t = 0.1$, generated by: (i) GS4-1($\rho_\infty, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty, \rho^s_\infty = \rho_\infty$) i.e., the case without the selective control feature, for $\rho_\infty = 0.8$ and 0.7



(a) (b)

Figure 5.40: Plot of T and $\dot{T}$ (for the convective heat transfer problem with thermal shock, Section 5.4) as a function of x at $t = 1$ using $\Delta t = 0.1$, generated by: (i) GS4-1($\rho_\infty = 0.6, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = \rho^s_\infty = 0.6$) i.e., the case without the selective control feature

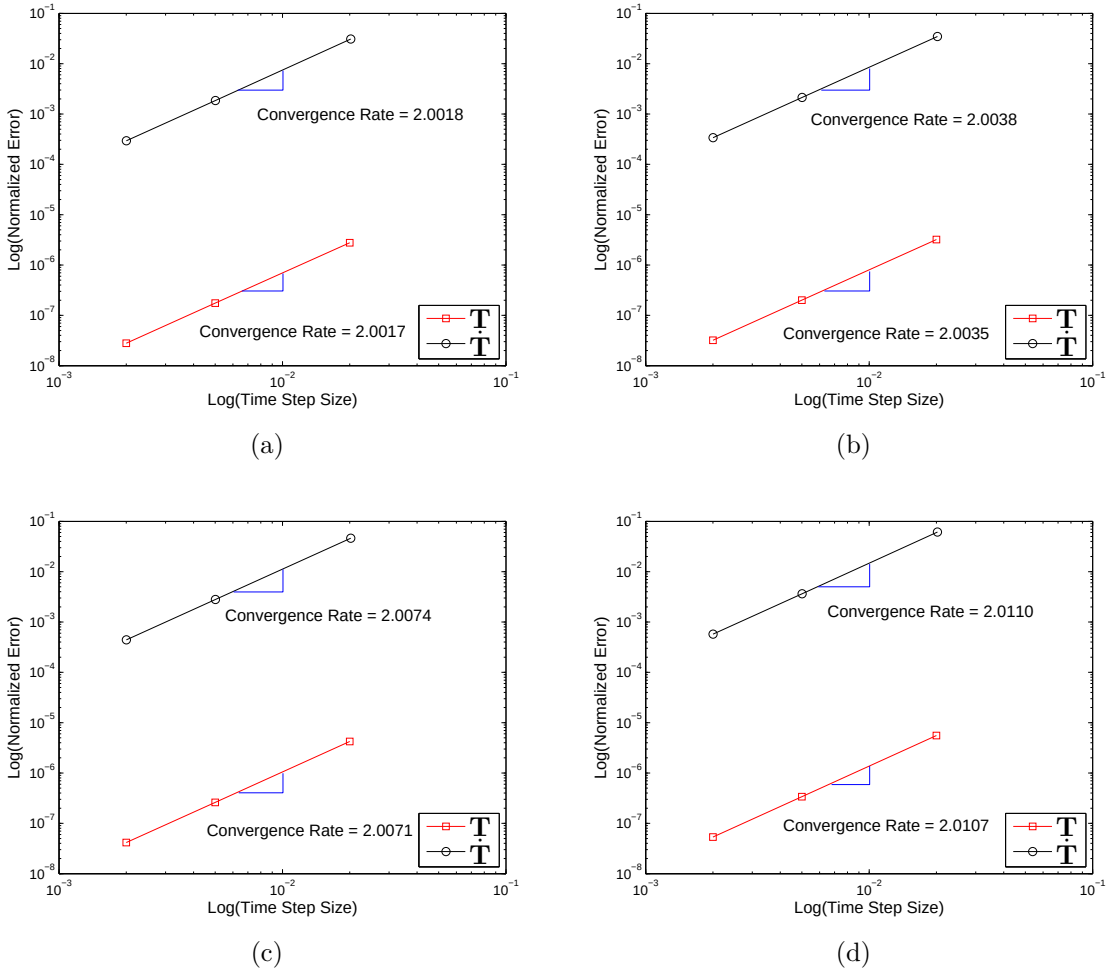


Figure 5.41: Convergence plot (for the convective heat transfer problem with thermal shock, Section 5.4) of the GS4-1 framework with: (a) $\rho_\infty = 1$, $\rho_\infty^s = 0$, (b) $\rho_\infty = 0.9$, $\rho_\infty^s = 0$, (c) $\rho_\infty = 0.7$, $\rho_\infty^s = 0$, and (d) $\rho_\infty = 0.6$, $\rho_\infty^s = 0$, utilizing the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$

Chapter 6

Extension of GS4-1 Framework to Nonlinear First Order Transient Systems

6.1 Introduction

The presence of nonlinearities in dynamic problems for most engineering applications has long been recognized to cause a demand for robust computational methods for obtaining stable and acceptable numerical solutions. In the previous chapter, we have primarily focused on the application of the newly developed GS4-1 framework to linear first order transient systems. Alternatively, in this chapter, we wish to tackle the challenges of applications pertaining to nonlinear first order transient systems that frequently arise in fluid dynamics problems. Here, our primary objective is to show how the GS4-1 framework, originally developed for linear first order transient system, can be properly extended for nonlinear applications. Additionally, using a number of numerical examples, we consistently demonstrate the abilities of the GS4-1 framework for such applications as those seen in the linear dynamic counterparts; i.e., whilst on one hand we show that an equal amount of numerical dissipation (i.e, $\rho_\infty = \rho_\infty^s$) as in the existing methods leads to non-physical instability in the time derivative variable for a minimal numerical dissipation required to obtain acceptable solution of the primary variable, on the other hand, we demonstrate how this instability can easily be turned off via the

selective control feature (i.e, $\rho_\infty \neq \rho_\infty^s$). Not only that this new feature enables the attainment of physically representative solutions of both variables, it also provides the necessary avenue to avoid imposing over-dissipation (i.e., imposing greater amount of numerical dissipation than is needed) to the variables. More importantly, we demonstrate in this chapter how imposing over-dissipation may yield numerical solution that, although is stable, deviates from the actual dynamics of the problem; hence demonstrate the crucial need to avoid such an issue, a requirement that can easily be achieved by the new selective control feature. For completeness, we also demonstrate the order of accuracy of the algorithms in the new framework when properly applied to nonlinear applications. The analysis presented in this chapter, therefore, further and consistently demonstrates the ability and advantages of the newly developed GS4-1 framework in comparison to the existing methods.

Turning attention to issues and challenges related to solving nonlinear dynamic problems, typical procedures that are routinely employed often take the linear algorithms (referred to as the "*basic primitive algorithm*") and simply apply an iterative process such as the Newton-Raphson method to account for the nonlinear effects. How the nonlinearity of the problem can be properly taken into account in the Newton-Raphson iteration is subject to scrutiny to ensure physically representative solutions as well as convergence of the nonlinear iterations and completion of the entire simulation. Therefore, such an important issue will be addressed in this chapter. One way of doing it is by employing the classical time weighted residual approach [69] that was originally used when designing the GS4-1 framework for linear cases. One important question that arises is whether this classical approach can readily serve the purpose of extending the developed GS4-1 framework to nonlinear dynamic situations such that the targeted objectives are met, or, do we need to look into other new alternatives that can provide a sound theoretical basis for extending the basic primitive algorithm (i.e., the algorithms for the linear cases) to nonlinear dynamic applications? Toward this end, we wish to put forth similar theoretical ideas that we have developed in the past for nonlinear structural dynamics systems (second order dynamic systems); that is, we employ the new normalized time weighted residual approach for applicability in nonlinear first order transient systems. To understand the importance of and the contribution of such a

theoretical approach, we next briefly review and highlight the essential concepts.

We have previously presented a new theoretical idea, termed as the *normalized time weighted residual approach*, to accurately model nonlinear structural dynamic problems (second order dynamic systems) [41–44]. The accurate treatment of the nonlinear terms present in the governing equation can be accomplished via the theoretical features emanating from such an approach, unlike the classical time weighted residual approach. The key idea behind this new approach, in comparison to the classical counterpart, was to provide the necessary avenue to individually weigh and normalize each term in the nonlinear semi-discretized equations of motion which consequently enables the nonlinear terms in the equation of motion to be treated specifically, allowing for more appropriate treatments of such terms. More noteworthy is the fact that the normalized time weighted residual approach can lead to all possible ways of how the nonlinear terms can be accurately treated. In general, for any nonlinear problems, the number of distinct nonlinear treatments that can be derived from the normalized time weighted residual approach depends on the degree of nonlinearity of the problem. As seen in our previous studies [41–44], employing such approach for structural dynamics problem (second order dynamic system) with Saint Vernant-Kirchhoff material model yields two different nonlinear treatments, both of which have significant advantages over the classical approach. Due to this achievement, we wish to employ the normalized time weighted residual approach for solving the nonlinear first order systems considered in this chapter. Furthermore, this approach also naturally and consistently reverts to the classical counterpart when applied to linear dynamic situations; hence, it is the proper way to handle nonlinear dynamic situations.

While our previous works involving the normalized time weighted residual approach were focused on applications in structural dynamic problems (second order dynamic systems), in this study we wish to employ the same conceptual approach to provide accurate treatment of the nonlinear terms that may arise in applications pertaining to nonlinear first order transient systems. The description of how this can be accomplished is presented in this chapter where we discuss the extension of the newly developed GS4-1 framework from applications in linear first order transient systems

to their nonlinear counterparts, with illustrations to radiation heat transfer and simplified fluid dynamic problems.

6.2 Radiation Heat Transfer

6.2.1 Governing Equation

To describe how the present GS4-1 framework, originally developed for linear first order transient systems [66] (see also [67]), can properly be extended for use in nonlinear problems, we shall consider from the start an application to a transient radiation heat transfer problem. For ease of explanation, we consider for this particular problem a Single-Degree-Of-Freedom (SDOF) example to reduce the complexity of such a nonlinear problem (by eliminating the need for spatial discretization). Such a SDOF example represents the radiation heat transfer occurring in an isothermal body (i.e., having uniform temperature). In this way, the discussions here can be focused solely on how the GS4-1 framework can be used to solve the problem, and in particular, how the normalized time weighted residual approach can be employed to properly treat the nonlinear term involved in the problem. In this regard, the present example is of great interest due to its high degree of nonlinearity, which allows for a complete description of all the possible approaches (of how the nonlinear term can be treated according to the normalized time weighted residual approach) that can be employed in the computational procedure (this will be described in detailed in Sections 6.2.2, 6.2.3 and 6.2.4). We next present the governing equation of interest.

For surfaces at high temperature T being exposed to a surrounding medium with lower temperature T_∞, radiation is emitted by the matter and it may be attributed to changes in the electric configuration of the constituent atoms or molecules. The radiative heat transfer loss (q'') can be calculated according to

$$q'' = \sigma\epsilon(T^4 - T_\infty^4) \tag{6.2.1}$$

where σ is the Stefan-Boltzmann's constant and ϵ is the emissivity of the surface. When the problem is time-dependent, we have the following governing equation

$$\dot{T} + \sigma\epsilon(T^4 - T_\infty^4) = 0 \tag{6.2.2}$$

with initial temperature

$$T(t = 0) = T_0 \tag{6.2.3}$$

As seen in equation (6.2.2), this illustrative problem is highly nonlinear due to the presence of the nonlinear term T^4. Therefore, although being only SDOF, solving equation (6.2.2) is a non-trivial matter. Obtaining physically representative and accurate solutions, and in some cases, the convergence of the nonlinear iterations within a time step require an accurate treatment of the nonlinear term in the computational procedure. This is described next.

6.2.2 Classical Time Weighted Residual Approach

As previously mentioned, one way the nonlinear term can be treated is by employing the classical time weighted residual approach [69], that was originally used in developing the GS4-1 framework for linear cases as described in Chapter 3. Representing the ordinary differential equation considered in this section (equation (6.2.2)) using such an approach with an arbitrary weighted time field W requires the following

$$\int_0^{\Delta t} W[\dot{T} + \sigma\epsilon P - F]dt = 0 \tag{6.2.4}$$

where W is assumed, as previously, to be a degenerated scalar polynomial function of the form (see Reference [29] for details)

$$W = 1 + w_1\Gamma + w_2\Gamma^2 \tag{6.2.5}$$

where $\Gamma = \frac{t}{\Delta t}$ and $t \in [0, \Delta t]$. Meanwhile, the terms w_1 and w_2 are intermediate parameters given by equation (3.3.86).

In equation (6.2.4), the term P represents the nonlinear term (T^4) in equation (6.2.2).

It is worthy to note here that using the classical time weighted residual approach, the whole nonlinear term is represented by another single variable, denoted as P. Meanwhile, $F = \sigma\epsilon T_\infty^4$. The time derivative variable ($\dot{T}$) and the nonlinear term (P) in equation (6.2.4) are then approximated in the same manner as done when deriving the GS4-1 framework for linear cases (see Chapter 3), i.e., by using asymptotic series expansions, whereas the load term (F) is approximated using Taylor series expansion as follows

$$\dot{T} = \dot{T}_n + \Lambda_6 \frac{\Delta \dot{T}}{\Delta t} t \tag{6.2.6}$$

$$P = P_n + \Lambda_4 \dot{P}_n t + \Lambda_5 \frac{\Delta \dot{P}}{\Delta t} t^2 \tag{6.2.7}$$

$$F = F_n + \frac{F_{n+1} - F_n}{\Delta t} t \tag{6.2.8}$$

where

$$\Delta \dot{T} = \dot{T}_{n+1} - \dot{T}_n \tag{6.2.9}$$

$$\Delta \dot{P} = \dot{P}_{n+1} - \dot{P}_n \tag{6.2.10}$$

and, in general, for any time level

$$P = P(T) = T^4 \tag{6.2.11}$$

$$\dot{P} = \frac{\partial}{\partial t} P(T) = 4T^3 \dot{T} \tag{6.2.12}$$

Recall from Chapter 3 that the notations and labelling of the parameters Λ_4, Λ_5, and Λ_6 introduced above are chosen as such to ensure consistency of the new GS4-1 framework with respect to the existing GS4-2 framework allowing for the development of an isochronous integration framework that can be used to solve both first and second order dynamic systems without having to resort to the individual framework (this will become more apparent in Chapter 7 where we present such a development).

Substituting the approximations, equations (6.2.6) to (6.2.8), into equation (6.2.4), dividing the resulting equation by $\int_0^{\Delta t} W dt$ and defining for convenience (as done previously in Chapter 3, hence the consistency)

$$W_i = \frac{\int_0^{\Delta t} W(\frac{t}{\Delta t})^i \, dt}{\int_0^{\Delta t} W dt} \tag{6.2.13}$$

yields the following temporally discrete governing equation

$$\dot{T}_n + \Lambda_6 W_1 \Delta\dot{T} + \sigma\epsilon(P_n + \Lambda_4 W_1 \Delta t \dot{P}_n + \Lambda_5 W_2 \Delta t \Delta\dot{P}) = F_n + W_1(F_{n+1} - F_n) \tag{6.2.14}$$

The above formulation can be represented as

$$\widetilde{\dot{T}} + \sigma\epsilon\widetilde{P} + = \widetilde{F} \tag{6.2.15}$$

where

$$\widetilde{\dot{T}} = \dot{T}_n + \Lambda_6 W_1 \Delta\dot{T} \tag{6.2.16}$$

$$\widetilde{P} = P_n + \Lambda_4 W_1 \Delta t \dot{P}_n + \Lambda_5 W_2 \Delta t \Delta\dot{P} \tag{6.2.17}$$

$$\widetilde{F} = F_n + W_1(F_{n+1} - F_n) \tag{6.2.18}$$

The associated expressions for the updates of the variables at the end of each time level are chosen as (adapting those for the linear cases, see Chapter 3)

$$T_{n+1} = T_n + \lambda_4 \dot{T}_n \Delta t + \lambda_5 \Delta\dot{T} \Delta t \tag{6.2.19}$$

$$P_{n+1} = P_n + \lambda_4 \dot{P}_n \Delta t + \lambda_5 \Delta\dot{P} \Delta t \tag{6.2.20}$$

$$\dot{T}_{n+1} = \dot{T}_n + \Delta\dot{T} \tag{6.2.21}$$

$$\dot{P}_{n+1} = \dot{P}_n + \Delta\dot{P} \tag{6.2.22}$$

At this point, it is worth noting that the above formulation involves the same

algorithmic parameters, namely the terms $\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5, that appear in the derivations of the original GS4-1 framework (i.e., for linear cases) described in Chapter 3. One important point to recall here is that in solving nonlinear problems, typical procedures that are routinely applied often use the algorithms for linear cases as the basic primitive algorithm to march the solutions in time, and simply apply an iterative process such as the Newton-Raphson method to account for the nonlinear dynamic effects. This can be achieved by approximating the variables and expressing the updates of these variables in the same manner as done for the linear cases (as shown above). This allows the computational procedure for solving the nonlinear problems to employ the original GS4-1 framework (described and employed previously in Chapter 3 to 5 for linear cases) as the basic primitive algorithm to march the solutions in time. How the nonlinearity of the problem is taken into account in the computational procedure depends on the nonlinear approach employed. In particular, the above formulation is the description of how the nonlinear term is treated by the classical time weighted residual approach. Other/alternative treatments for the nonlinear term are also available, as described later in Section 6.2.3. The differences between these nonlinear treatments in the computational procedure will become apparent in Section 6.2.4 where we present the complete computational procedures for these approaches to solve the nonlinear problem of interest.

Accordingly, from Chapter 3, the above algorithmic parameters can be expressed in terms of the principal root (ρ_∞) and the spurious root (ρ_∞^s) as follows

$$\begin{aligned}
&\Lambda_6 W_1 = \frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)(1+\rho_\infty^s)}, \quad \Lambda_5 W_2 = \frac{1}{(1+\rho_\infty)(1+\rho_\infty^s)}, \\
&\Lambda_4 W_1 = \frac{1}{1+\rho_\infty}, \quad W_1 = \frac{1}{1+\rho_\infty}, \quad \lambda_4 = 1, \quad \lambda_5 = \frac{1}{1+\rho_\infty^s}
\end{aligned} \tag{6.2.23}$$

Upon rearrangements, equations (6.2.16) to (6.2.21) can also be represented in another form as follows (note that in the computational procedures described in Section 6.2.4, we may for convenience choose to use the equations from either of these representations)

$$\widetilde{\dot{T}} = \left(1 - \frac{\lambda_4}{\lambda_5}\Lambda_6 W_1\right)\dot{T}_n + \frac{\Lambda_6 W_1}{\lambda_5}\frac{\Delta T}{\Delta t} \tag{6.2.24}$$

$$\widetilde{P} = P_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta P \tag{6.2.25}$$

where

$$\Delta P = P_{n+1} - P_n \tag{6.2.26}$$

$$P_n = P\left(T_n\right) = T_n^4 \tag{6.2.27}$$

$$P_{n+1} = P\left(T_{n+1}\right) = T_{n+1}^4 \tag{6.2.28}$$

with the corresponding expressions for the update of the variables at the end of each time level as follows

$$T_{n+1} = T_n + \Delta T \tag{6.2.29}$$

$$\dot{T}_{n+1} = \frac{1}{\lambda_5}\frac{\Delta T}{\Delta t} + \left(1 - \frac{\lambda_4}{\lambda_5}\right)\dot{T}_n \tag{6.2.30}$$

6.2.3 Normalized Time Weighted Residual Approach

We now proceed in this section to describe how the nonlinear term can be treated by other nonlinear treatments emanating from the normalized time weighted residual approach. For this purpose, we first review the concept of such a new approach that has previously been shown to provide an improvement for second order nonlinear dynamic systems [41–44].

The first necessary step in the normalized time weighted residual approach is to represent the nonlinear ordinary differential equation[1] in all different possible forms.

[1] semi-discretized if the problem is spatially dependent

For the present example, equation (6.2.2) can be represented in four different forms as follows

1. $\dot{T} + \sigma\epsilon T^4 = F$ (i.e., the original form)
2. $\dot{T} + \sigma\epsilon\zeta T = F$ where $\zeta = T^3$
3. $\dot{T} + \sigma\epsilon\zeta^2 = F$ where $\zeta = T^2$
4. $\dot{T} + \sigma\epsilon\zeta T^2 = F$ where $\zeta = T^2$

where $F = \sigma\epsilon T_\infty^4$. At a first glance, the above four forms seem to yield the same equation (i.e., equation (6.2.2)). Whilst that is true, it is worth to carefully notice that in the above four forms, the nonlinear term (T^4) is represented differently for each form, i.e.,

- for (1), the nonlinear term is kept at its original form, T^4
- for (2), the nonlinear term is represented by the product of ζ and T where $\zeta = T^3$
- for (3), the nonlinear term is represented by the square of ζ where $\zeta = T^2$
- for (4), the nonlinear term is represented by the product of ζ and T^2 where $\zeta = T^2$

These different representations of the nonlinear term will lead to all the possible different nonlinear treatments according to the normalized time weighted residual approach as described in detailed in the following section.

For each representation, the normalized time weighted residual approach is then employed as follows [41–44]. Each term (denoted in general by χ_i, for example $\chi_i = \dot{T}, T, \zeta, F$ etc) appearing in such a representation is weighted individually by using the time weighted function W. Additionally, each term is also normalized individually by using the same time weighted function. This is illustrated in general form as follows:

$$\frac{\int_0^{\Delta t} W \chi_i \, dt}{\int_0^{\Delta t} W dt} \tag{6.2.31}$$

If χ_i is a constant, then

$$\frac{\int_0^{\Delta t} W \chi_i \, dt}{\int_0^{\Delta t} W dt} = \chi_i \tag{6.2.32}$$

Such a theoretical idea provides the necessary avenue to represent the nonlinear ordinary differential equation by a time weighted residual representation in order to discretize the equation in time.

The next step in the normalized time weighted residual approach is to approximate and update the variables involved in the representation in the same manner as done previously for the linear cases. This, hence, allows the original GS4-1 framework to be used as the basic primitive algorithm to march the solutions in time. Note that this particular step is similar to that for the classical time weighted residual approach described previously in Section 6.2.2.

We now illustrate this procedure for each of the four different forms listed above (all of which represent the considered radiation heat transfer problem), which leads to the four different nonlinear treatments for such a problem. Meanwhile, the complete computational procedures for all these nonlinear treatments will be presented in Section 6.2.4.

6.2.3.1 Nonlinear Treatment 1

For this first option, we recall that the nonlinear ordinary differential equation for the radiation heat transfer problem can be represented in its original form as

$$\dot{T} + \sigma \epsilon T^4 = F \tag{6.2.33}$$

where

$$F = \sigma \epsilon T_{\infty}^4 \tag{6.2.34}$$

We next employ the normalized time weighted residual approach described in Section 6.2.3 to equation (6.2.33) as follows

$$\frac{\int_0^{\Delta t} W\dot{T}dt}{\int_0^{\Delta t} Wdt} + \frac{\int_0^{\Delta t} W\sigma dt}{\int_0^{\Delta t} Wdt}\frac{\int_0^{\Delta t} W\epsilon dt}{\int_0^{\Delta t} Wdt}\frac{\int_0^{\Delta t} WTdt}{\int_0^{\Delta t} Wdt}\frac{\int_0^{\Delta t} WTdt}{\int_0^{\Delta t} Wdt}\frac{\int_0^{\Delta t} WTdt}{\int_0^{\Delta t} Wdt}\frac{\int_0^{\Delta t} WTdt}{\int_0^{\Delta t} Wdt}$$
$$- \frac{\int_0^{\Delta t} WFdt}{\int_0^{\Delta t} Wdt} = 0 \tag{6.2.35}$$

which can be simplified (using equation (6.2.32)) and subsequently represented in a more convenient form as follows

$$\frac{\int_0^{\Delta t} W\dot{T}dt}{\int_0^{\Delta t} Wdt} + \sigma\epsilon\left(\frac{\int_0^{\Delta t} WTdt}{\int_0^{\Delta t} Wdt}\right)^4 - \frac{\int_0^{\Delta t} WFdt}{\int_0^{\Delta t} Wdt} = 0 \tag{6.2.36}$$

We next approximate the variables $(\dot{T}, T, F)$ appearing in equation (6.2.36) in the same manner as done previously when deriving the original GS4-1 framework (for linear cases, described in Chapter 3), i.e.,

$$\dot{T} = \dot{T}_n + \Lambda_6 \frac{\Delta \dot{T}}{\Delta t} t \tag{6.2.37}$$

$$T = T_n + \Lambda_4 \dot{T}_n t + \Lambda_5 \frac{\Delta \dot{T}}{\Delta t} t^2 \tag{6.2.38}$$

$$F = F_n + \frac{F_{n+1} - F_n}{\Delta t} t \tag{6.2.39}$$

Substituting the above approximations into equation (6.2.36) and making use of equation (6.2.13), equation (6.2.36) becomes

$$\widetilde{\dot{T}} + \sigma\epsilon\widetilde{T}^4 = \widetilde{F} \tag{6.2.40}$$

where

$$\widetilde{\dot{T}} = \dot{T}_n + \Lambda_6 W_1 \Delta \dot{T} \tag{6.2.41}$$

$$\widetilde{T} = T_n + \Lambda_4 W_1 \Delta t \dot{T}_n + \Lambda_5 W_2 \Delta t \Delta \dot{T} = T_n + \frac{\Lambda_5 W_2}{\lambda_5} \Delta T \tag{6.2.42}$$

$$\widetilde{F} = F_n + W_1 (F_{n+1} - F_n) \tag{6.2.43}$$

and with the same expressions for the updates of the variables at the end of each time level.

6.2.3.2 Nonlinear Treatment 2

In this alternative treatment, we represent the nonlinear ordinary differential equation of the radiation heat transfer problem (equation (6.2.2)) as follows

$$\dot{T} + \sigma \epsilon \zeta T = F \tag{6.2.44}$$

where $\zeta = T^3$ (and $F = \sigma \epsilon T_\infty^4$ as before). We next employ the normalized time weighted residual approach to equation (6.2.44) and simplify to yield

$$\frac{\int_0^{\Delta t} W \dot{T} dt}{\int_0^{\Delta t} W dt} + \sigma \epsilon \frac{\int_0^{\Delta t} W \zeta dt}{\int_0^{\Delta t} W dt} \frac{\int_0^{\Delta t} W T dt}{\int_0^{\Delta t} W dt} - \frac{\int_0^{\Delta t} W F dt}{\int_0^{\Delta t} W dt} = 0 \tag{6.2.45}$$

We proceed next by approximating the variables $(\dot{T}, \zeta, T, F)$ appearing in equation (6.2.45) in the same manner as done for the other approaches

$$\dot{T} = \dot{T}_n + \Lambda_6 \frac{\Delta \dot{T}}{\Delta t} t \tag{6.2.46}$$

$$\zeta = \zeta_n + \Lambda_4 \dot{\zeta}_n t + \Lambda_5 \frac{\Delta \dot{\zeta}}{\Delta t} t^2 \tag{6.2.47}$$

$$T = T_n + \Lambda_4 \dot{T}_n t + \Lambda_5 \frac{\Delta \dot{T}}{\Delta t} t^2 \tag{6.2.48}$$

$$F = F_n + \frac{F_{n+1} - F_n}{\Delta t} t \tag{6.2.49}$$

with the associated updates

$$T_{n+1} = T_n + \lambda_4 \dot{T}_n \Delta t + \lambda_5 \Delta \dot{T} \Delta t \tag{6.2.50}$$

$$\zeta_{n+1} = \zeta_n + \lambda_4 \dot{\zeta}_n \Delta t + \lambda_5 \Delta \dot{\zeta} \Delta t \tag{6.2.51}$$

$$\dot{T}_{n+1} = \dot{T}_n + \Delta \dot{T} \tag{6.2.52}$$

$$\dot{\zeta}_{n+1} = \dot{\zeta}_n + \Delta \dot{\zeta} \tag{6.2.53}$$

As previously highlighted, it is to be noted that in this treatment, the nonlinear term is represented as a product of ζ and T, where ζ represents T^3. However, as compared to the first nonlinear treatment of this normalized time weighted residual approach (Section 6.2.3.1) where the nonlinear term is represented by T^4 (i.e., the original form), in this second treatment both ζ and T are approximated and updated independently as shown above. This yields a different nonlinear treatment (as here ζ represents T^3, so we are actually approximating and updating ' T^3 ' and ' T ' independently).

Upon substitution of the above approximations into equation (6.2.45) and with the use of equation (6.2.13), equation (6.2.45) can now be represented as follows

$$\widetilde{\dot{T}} + \sigma \epsilon \widetilde{\zeta} \widetilde{T} = \widetilde{F} \tag{6.2.54}$$

where

$$\widetilde{\dot{T}} = \dot{T}_n + \Lambda_6 W_1 \Delta\dot{T} \tag{6.2.55}$$

$$\widetilde{\zeta} = \zeta_n + \Lambda_4 W_1 \Delta t \dot{\zeta}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\zeta} \tag{6.2.56}$$

$$\widetilde{T} = T_n + \Lambda_4 W_1 \Delta t \dot{T}_n + \Lambda_5 W_2 \Delta t \Delta\dot{T} \tag{6.2.57}$$

$$\widetilde{F} = F_n + W_1 (F_{n+1} - F_n) \tag{6.2.58}$$

Using equations (6.2.51) and (6.2.56), we can express $\widetilde{\zeta}$ in a more convenient form as follows

$$\widetilde{\zeta} = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5} \Delta\zeta \tag{6.2.59}$$

where

$$\Delta\zeta = \zeta_{n+1} - \zeta_n, \quad \zeta_{n+1} = T_{n+1}^3, \quad \zeta_n = T_n^3 \tag{6.2.60}$$

6.2.3.3 Nonlinear Treatment 3

Alternatively, we can represent the considered nonlinear ordinary differential equation (equation (6.2.2)) in the following form

$$\dot{T} + \sigma\epsilon\zeta^2 = F \tag{6.2.61}$$

where, for this option, $\zeta = T^2$. Employing the normalized time weighted residual approach to equation (6.2.61), and further simplification (from use of equation (6.2.32)) yields

$$\frac{\int_0^{\Delta t} W\dot{T} dt}{\int_0^{\Delta t} W dt} + \sigma\epsilon \left(\frac{\int_0^{\Delta t} W\zeta dt}{\int_0^{\Delta t} W dt} \right)^2 - \frac{\int_0^{\Delta t} WF dt}{\int_0^{\Delta t} W dt} = 0 \tag{6.2.62}$$

Approximating the variables $(\dot{T}, \zeta, F)$ appearing in equation (6.2.62) in the same manner as done for the other treatments (as described before), and substituting these

approximations to equation (6.2.62) yields the following algorithmic equation (with the use of equation (6.2.13))

$$\widetilde{\dot{T}} + \sigma\epsilon\widetilde{\zeta}^{\,2} = \widetilde{F} \tag{6.2.63}$$

where

$$\widetilde{\dot{T}} = \dot{T}_n + \Lambda_6 W_1 \Delta\dot{T} \tag{6.2.64}$$

$$\widetilde{\zeta} = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta\zeta \tag{6.2.65}$$

$$\widetilde{F} = F_n + W_1(F_{n+1} - F_n) \tag{6.2.66}$$

while, for this option,

$$\Delta\zeta = \zeta_{n+1} - \zeta_n, \quad \zeta_{n+1} = T_{n+1}^2, \quad \zeta_n = T_n^2 \tag{6.2.67}$$

Again, it is easy to note that the resulting algorithmic equation in this option (equation (6.2.63)) is different from that for the other treatments due to the different representation of the nonlinear term, which leads to the different approximation and update and subsequently different algorithmic equation to be solved in the computational procedure.

6.2.3.4 Nonlinear Treatment 4

Finally, we describe the last option that can be derived from the normalized time weighted residual approach. For this purpose, we first express the nonlinear ordinary differential equation of interest (equation (6.2.2)) in the following form

$$\dot{T} + \sigma\epsilon\zeta T^2 = F \tag{6.2.68}$$

where, again, $\zeta = T^2$ (it is to be noted here that, although ζ in this treatment also

represents T^2 as in the previous treatment, the ordinary differential equation for this option is not expressed in the same manner as that for the previous one (see equation (6.2.61))). From the use of the normalized time weighted residual approach and further simplification yields

$$\frac{\int_0^{\Delta t} W\dot{T}dt}{\int_0^{\Delta t} Wdt} + \sigma\epsilon\frac{\int_0^{\Delta t} W\zeta dt}{\int_0^{\Delta t} Wdt}\left(\frac{\int_0^{\Delta t} WTdt}{\int_0^{\Delta t} Wdt}\right)^2 - \frac{\int_0^{\Delta t} WFdt}{\int_0^{\Delta t} Wdt} = 0 \tag{6.2.69}$$

Approximating the variables $(\dot{T}, \zeta, T, F)$ appearing in equation (6.2.69) in the same manner as done for the other treatments, substituting these approximations to equation (6.2.69), and making use of equation (6.2.13) yield

$$\widetilde{\dot{T}} + \sigma\epsilon\widetilde{\zeta}\widetilde{T}^2 = \widetilde{F} \tag{6.2.70}$$

where

$$\widetilde{\dot{T}} = \dot{T}_n + \Lambda_6 W_1 \Delta\dot{T} \tag{6.2.71}$$

$$\widetilde{\zeta} = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta\zeta \tag{6.2.72}$$

$$\widetilde{T} = T_n + \Lambda_4 W_1 \Delta t\dot{T}_n + \Lambda_5 W_2 \Delta t\Delta\dot{T} \tag{6.2.73}$$

$$\widetilde{F} = F_n + W_1(F_{n+1} - F_n) \tag{6.2.74}$$

where, as in the previous treatment,

$$\Delta\zeta = \zeta_{n+1} - \zeta_n, \quad \zeta_{n+1} = T_{n+1}^2, \quad \zeta_n = T_n^2 \tag{6.2.75}$$

Remark 6.2.1

For any nonlinear dynamic problems, the number of distinct nonlinear treatments that can be derived from the normalized time weighted residual approach depends on the degree of nonlinearity of the problem:

- *For the particular nonlinear problem considered in this section, the normalized*

time weighted residual approach yields four different nonlinear treatments as described above. This is due to the fact that the nonlinear term is to the fourth power of the primary variable.

- *When the nonlinear term is to the third power of the primary variable, this approach yields two different nonlinear treatments (see [41–44]).*
- *Meanwhile, when the nonlinear term is only to the second power of the primary variable, this approach yields only one nonlinear treatment (see Section 6.3).*

Consistently, when the problem is linear, this approach reverts to the classical counterpart.

6.2.4 Computational Procedures for All Approaches

We have previously described and presented in Sections 6.2.2 and 6.2.3 the different time weighted residual approaches to discretize the governing equation of the radiation heat transfer problem which is nonlinear. These include the classical time weighted residual approach (Section 6.2.2) and the four distinct nonlinear treatments emanating from the normalized time weighted residual approach (Section 6.2.3). From a close observation of these different nonlinear treatments, the underlying differences between them lie in the approximation of the nonlinear term (denoted in general for all treatments as P), and subsequently, in the algorithmic representation of this nonlinear term ($\widetilde{P}$) in the resulting algorithmic equation. It is also to be noted that the above descriptions for all approaches are to be completed by the description of an iterative process such as the Newton-Raphson method to obtain the full description of the computational procedures needed to solve the problem. In this section, we present such a description (i.e., the iterative process) and for this purpose, we employ the Newton-Raphson method. To improve the clarity of the following description, we present these computational procedures in a general form that is common to all approaches described above, and additionally, we mention the differences in the approaches as they arise.

In short, in the computational procedures to be described next, the distinct approaches will have different definitions of: (1) the algorithmic nonlinear term (i.e.,

$\widetilde{P}$) involved in the calculation of the residual at each nonlinear iteration within one time level, and (2) the corresponding tangential stiffness ($= \frac{\partial \widetilde{P}}{\partial \widetilde{T}}$) involved in the calculation of the jacobian at each nonlinear iteration within one time level. The different definitions of these two terms for all aproaches will be made clear in the following description of the computational procedures. Unless otherwise stated, all the other steps involved hold for all approaches. The complete description of the computational procedures for solving the nonlinear problem of interest follows next.

Algorithm 5
Computational Procedures (All Approaches) for Solving the Radiation Heat Transfer Problem, Section 6.2

Given/knowing the solutions at previous time level t_n (i.e., T_n and $\dot{T}_n$), we seek the solutions of the primary variable and its time derivative at the next time level t_{n+1} (i.e., T_{n+1} and $\dot{T}_{n+1}$). At the beginning of t_{n+1} time level, we initially predict the solutions using known values at previous time level as follows [2]

$$\begin{aligned} T^k_{n+1} &= T_n \\ \dot{T}^k_{n+1} &= \left(1 - \frac{\lambda_4}{\lambda_5}\right) \dot{T}_n \end{aligned} \tag{6.2.76}$$

where the superscript k is the nonlinear iteration counter. We then calculate the algorithmic variables $\widetilde{T}^k$ and $\widetilde{\dot{T}}^k$ in terms of the predicted values T^k_{n+1} and $\dot{T}^k_{n+1}$ as follows

$$\widetilde{T}^k = T_n + \frac{\Lambda_5 W_2}{\lambda_5}(T^k_{n+1} - T_n) \tag{6.2.77}$$

$$\widetilde{\dot{T}}^k = \dot{T}_n + \Lambda_6 W_1(\dot{T}^k_{n+1} - \dot{T}_n) \tag{6.2.78}$$

while the algorithmic load term $\widetilde{F}^k$ is determined from its known values at t_n and

[2] the latter equation is taken from equation (6.2.30)

t_{n+1} *as follows*

$$\widetilde{F}^k = F_n + W_1(F_{n+1} - F_n) \tag{6.2.79}$$

We next enter the nonlinear iteration loop. This is done as follows:

(i) In this nonlinear iteration loop, we first calculate the residual resulting from using the predicted algorithmic variables at the k-th iteration. The residual at the k-th iteration $\widetilde{R}^k$ *can be represented in a general form that is common to all approaches as follows*

$$\widetilde{R}^k = \widetilde{\dot{T}}^k + \sigma\epsilon\widetilde{P}^k - \widetilde{F}^k \tag{6.2.80}$$

where $\widetilde{P}^k$ *represents the algorithmic nonlinear term (in general form) at the k-th iteration. However, as previously mentioned and described in the preceeding section, the distinct approaches have different definitions of this term due to the different ways of how the actual nonlinear term is represented in these approaches as described before. This is summarized as follows:*

- *For the classical time weighted residual approach (Section 6.2.2), the algorithmic nonlinear term* ($\widetilde{P}^k$) *is given by equation* (6.2.25) *as follows*

$$\widetilde{P}^k = P_n + \frac{\Lambda_5 W_2}{\lambda_5}(P^k_{n+1} - P_n) \tag{6.2.81}$$

  where

$$P_n = (T_n)^4, \quad P^k_{n+1} = (T^k_{n+1})^4 \tag{6.2.82}$$

- *For the normalized time weighted residual approach with the first nonlinear treatment (Section 6.2.3.1), this term can be determined from equation* (6.2.40) *to be*

$$\widetilde{P}^k = (\widetilde{T}^k)^4 \tag{6.2.83}$$

- *Meanwhile, for the normalized time weighted residual approach with the second nonlinear treatment (Section 6.2.3.2), this term can be determined from equation* (6.2.54) *to be*

$$\widetilde{P}^k = \widetilde{\zeta}^k \widetilde{T}^k \tag{6.2.84}$$

where

$$\widetilde{\zeta}^k = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}(\zeta_{n+1}^k - \zeta_n), \quad \zeta_n = (T_n)^3, \quad \zeta_{n+1}^k = (T_{n+1}^k)^3 \tag{6.2.85}$$

- *For the third nonlinear treatment emanating from the normalized time weighted residual approach (Section 6.2.3.3), this term can be determined from equation* (6.2.63) *to be*

$$\widetilde{P}^k = (\widetilde{\zeta}^k)^2 \tag{6.2.86}$$

where

$$\widetilde{\zeta}^k = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}(\zeta_{n+1}^k - \zeta_n), \quad \zeta_n = (T_n)^2, \quad \zeta_{n+1}^k = (T_{n+1}^k)^2 \tag{6.2.87}$$

- *Finally, for the fourth/last nonlinear treatment emanating from the normalized time weighted residual approach (Section 6.2.3.4), this term can be determined from equation* (6.2.70) *to be*

$$\widetilde{P}^k = \widetilde{\zeta}^k (\widetilde{T}^k)^2 \tag{6.2.88}$$

where

$$\widetilde{\zeta}^k = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}(\zeta_{n+1}^k - \zeta_n), \quad \zeta_n = (T_n)^2, \quad \zeta_{n+1}^k = (T_{n+1}^k)^2 \tag{6.2.89}$$

(ii) We next linearize the residual using Taylor expansion, truncate after the linear term and set the residual to vanish. This step yields the following equation

$$\frac{\partial \widetilde{R}^k}{\partial \widetilde{T}^k}\Delta\widetilde{T} = -\widetilde{R}^k \tag{6.2.90}$$

where $\Delta\widetilde{T} = \widetilde{T}^{k+1} - \widetilde{T}^k$, *and* $\frac{\partial \widetilde{R}^k}{\partial \widetilde{T}^k}$ *is the Jacobian given by (from the use of equation* (6.2.80)*)*

$$\begin{aligned}\frac{\partial \widetilde{R}^k}{\partial \widetilde{T}^k} &= \frac{\partial \widetilde{\dot{T}}^k}{\partial \widetilde{T}^k} + \sigma\epsilon\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k} - \frac{\partial \widetilde{F}^k}{\partial \widetilde{T}^k} \\ &= \frac{\Lambda_6 W_1}{\Lambda_5 W_2 \Delta t} + \sigma\epsilon\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k}\end{aligned} \tag{6.2.91}$$

The term $\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k}$ *in equation* (6.2.91) *is the tangential stiffness. Since* $\widetilde{P}^k$ *is different for the distinct approaches as given by equations* (6.2.81), (6.2.83), (6.2.84), (6.2.86), *and* (6.2.88), *respectively, the corresponding tangential stiffness for each approach is also different. This is summarized as follows*

- *For the classical approach (Section 6.2.2), the tangential stiffness (*$\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k}$*) can be expressed (from the use of equation* (6.2.81)*) as follows*

$$\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k} = 4(T_{n+1}^k)^3 \tag{6.2.92}$$

- *For the normalized time weighted residual approach with the first nonlinear treatment (Section 6.2.3.1), this term is given by (from the use of equation* (6.2.83)*)*

$$\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k} = 4(\widetilde{T}^k)^3 \tag{6.2.93}$$

- *Meanwhile, for the normalized time weighted residual approach with the second nonlinear treatment (Section 6.2.3.2), this term is given by (from the use of equation* (6.2.84)*)*

$$\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k} = \widetilde{\zeta}^k + 3\widetilde{T}^k(T_{n+1}^k)^2 \tag{6.2.94}$$

where

$$\widetilde{\zeta}^k = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}(\zeta_{n+1}^k - \zeta_n), \quad \zeta_n = (T_n)^3, \quad \zeta_{n+1}^k = (T_{n+1}^k)^3 \tag{6.2.95}$$

- *For the third nonlinear treatment emanating from the normalized time weighted residual approach (Section 6.2.3.3), this term is given by (from the use of equation* (6.2.86)*)*

$$\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k} = 4\widetilde{\zeta}^k T_{n+1}^k \tag{6.2.96}$$

where

$$\widetilde{\zeta}^k = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}(\zeta_{n+1}^k - \zeta_n), \quad \zeta_n = (T_n)^2, \quad \zeta_{n+1}^k = (T_{n+1}^k)^2 \tag{6.2.97}$$

- *And finally, for the normalized time weighted residual approach with the fourth nonlinear treatment (Section 6.2.3.4), this term is given by (from the use of equation* (6.2.88)*)*

$$\frac{\partial \widetilde{P}^k}{\partial \widetilde{T}^k} = 2(\widetilde{T}^k)^2 T_{n+1}^k + 2\widetilde{\zeta}^k \widetilde{T}^k \tag{6.2.98}$$

where

$$\widetilde{\zeta}^k = \zeta_n + \frac{\Lambda_5 W_2}{\lambda_5}(\zeta_{n+1}^k - \zeta_n), \quad \zeta_n = (T_n)^2, \quad \zeta_{n+1}^k = (T_{n+1}^k)^2 \tag{6.2.99}$$

Equation (6.2.90) *can be rearranged to the following form*

$$\widetilde{T}^{k+1} = \left(\frac{\partial \widetilde{R}^k}{\partial \widetilde{T}^k}\right)^{-1} \left(-\widetilde{R}^k + \frac{\partial \widetilde{R}^k}{\partial \widetilde{T}^k}\widetilde{T}^k\right) \tag{6.2.100}$$

(iii) If the problem had been spatially dependent (i.e., initial boundary value problem), in the next step we impose the Dirichlet boundary condition (if any) to the respective nodes in equation (6.2.100)*. This can be done via the use of equation* (6.2.77)*, i.e.,*

by imposing that $\widetilde{T}^{k+1}$ *in equation* (6.2.100) *at the respective nodes (corresponding to the boundary) takes the following value*

$$\widetilde{T}^{k+1} = T_n + \frac{\Lambda_5 W_2}{\lambda_5}(T_{\Gamma n+1} - T_n) \tag{6.2.101}$$

where $T_{\Gamma n+1}$ *is the known prescribed temperature at the boundary* Γ *at time* t_{n+1} *(note that, however, the particular radiation heat transfer problem considered here is not spatially dependent, therefore, no boundary conditions need to be imposed).*

(iv) We then solve for $\widetilde{T}^{k+1}$ *from equation* (6.2.100) *after imposing the boundary conditions appropriately.*

(v) Subsequently, we correct the algorithmic time derivative variable as follows [3]

$$\widetilde{\dot{T}}^{k+1} = \left(1 - \frac{\lambda_4}{\lambda_5}\Lambda_6 W_1\right)\dot{T}_n + \frac{\Lambda_6 W_1}{\Lambda_5 W_2}\frac{(\widetilde{T}^{k+1} - T_n)}{\Delta t} \tag{6.2.102}$$

and also correct the predicted value of T *for the* t_{n+1} *time level as follows* [4]

$$T_{n+1}^{k+1} = T_n + \frac{\lambda_5}{\Lambda_5 W_2}(\widetilde{T}^{k+1} - T_n) \tag{6.2.103}$$

(vi) Upon obtaining the above algorithmic variables, we then check if convergence of the solution is achieved

$$|\widetilde{T}^{k+1} - \widetilde{T}^{k}| = tol \tag{6.2.104}$$

where $\widetilde{T}^{k}$ *is the value of the algorithmic primary variable at the previous nonlinear iteration (i.e., k-th iteration) and tol is the user-specified tolerance value (i.e., convergence criterion for the Newton-Raphson iteration).*

[3] equation (6.2.102) can be obtained from the use of equations (6.2.24) and (6.2.42)
[4] equation (6.2.103) can be obtained from equation (6.2.42)

We repeat the nonlinear iteration (i.e., steps (i) to (vi) described above) until the solutions are converged (i.e., until equation (6.2.104) *is satisfied). Upon convergence, we update the variables at the end of* t_{n+1} *time level as follows* [5]

$$
\begin{aligned}
T_{n+1} &= T_n + \frac{\lambda_5}{\Lambda_5 W_2}(\widetilde{T}^{k+1} - T_n) \\
\dot{T}_{n+1} &= \frac{(T_{n+1} - T_n)}{\lambda_5 \Delta t} + \left(1 - \frac{\lambda_4}{\lambda_5}\right) \dot{T}_n
\end{aligned}
\tag{6.2.105}
$$

6.2.5 Numerical Illustrations

In this section, we first demonstrate the performance of the different nonlinear approaches described in Sections 6.2.2 and 6.2.3 when solving the nonlinear problem of interest, i.e., the radiation heat transfer problem described in Section 6.2.1. As previously said, in solving nonlinear problems, typical procedures that are routinely applied often take algorithms for linear cases and simply apply an iterative process such as the Newton-Raphson method to account for the nonlinear dynamic effects. Therefore, we will employ the algorithms in the GS4-1 framework, that is originally developed for linear cases, as the basic primitive algorithm to march the solution in time. How the nonlinearity of the problem is taken into account in the computational procedure depends on the nonlinear approach employed, either the classical one or the distinct nonlinear treatments emanating from the normalized time weighted residual approach. For the particular nonlinear problem considered in this section, the complete and detailed computational procedures for all approaches are described in Section 6.2.4. Upon illustrating the performance of the different nonlinear approaches, we proceed by demonstrating the advantage and ability of the present development via the new selective control feature for this nonlinear problem, in contrast to the existing methods without such a feature.

To demonstrate the performance of the different nonlinear treatments, we solve the radiation heat transfer problem using all approaches via the computational procedures presented in Section 6.2.4. The problem parameters are chosen as

[5] the latter is taken from equation (6.2.30)

follow: $\sigma = 5.6703 \times 10^{-8} W/m^2/K^4$, $\epsilon = 0.64$ (typical of oxidized Iron), $T_\infty = 0\,K$, and $T_0 = 200\,K$, with a time step size (Δt) of 5s and an end time t_{end} of 50s. Meanwhile, the convergence criterion (tolerance) for the Newton-Raphson iteration is $tol = 1 \times 10^{-12}$.

The exact solution for the case with $T_\infty = 0\,K$ is given by (in unit of K. Please also note that in all plots, T is in unit of K)

$$T(t) = \frac{T_0}{(3\sigma\epsilon T_0^3 t + 1)^{1/3}} \tag{6.2.106}$$

To evaluate the performance of these different nonlinear approaches, additionally, we compute the errors generated by these approaches. The error is defined as

$$Error = \left| \frac{Numerical - Exact}{Exact} \right| \times 100\% \tag{6.2.107}$$

Following the guidelines given in Section 4.4, the basic primitive algorithm used to march the solutions in time is first chosen as the nondissipative algorithm, which can be recovered in the GS4-1 framework with particular choice of $\rho_\infty = \rho_\infty^s = 1$. The results of the primary variable (T) and its time derivative ($\dot{T}$) as well as the corresponding errors generated by the different nonlinear treatments are shown in Figure 6.1. Looking at the solutions of T and the corresponding errors (Figures 6.1(a) and 6.1(c)), we can see that the classical approach is not able to capture the solutions of T correctly, and yields large deviation from the exact solution. On the other hand, the four different nonlinear treatments derived from the normalized time weighted residual approach are able to yield solutions of T that approach the exact solution. In particular, we can see that the first nonlinear treatment of such approach (Section 6.2.3.1) yields the best results in this case. However, looking at the solutions of $\dot{T}$ and the corresponding errors (Figures 6.1(b) and 6.1(d)), the opposite trends than those seen for the solution of T are observed; the classical approach seems to perform the best in generating the solution of $\dot{T}$ and yields the least error. Although this seems to highlight the ability of the classical approach, it is important to recall that the solution of T generated by this approach greatly

deviates from the exact solution as seen in Figure 6.1(a). Therefore, the classical approach is not reliable in this case due to such an inconsistency. Meanwhile, the solutions of $\dot{T}$ generated by the nonlinear treatments derived from the normalized time weighted residual approach are oscillatory as seen in Figure 6.1(b) and yields increasing errors in $\dot{T}$ (see Figure 6.1(d)). However, we claim that such solution behaviours are due to the algorithmic property of the nondissipative algorithm that was used as the basic primitive algorithm to march the solutions in time. Due to the zero-damping property of such a totally nondissipative algorithm, the solutions of the time derivative variable are often oscillatory (as demonstrated in the previous chapter). In other words, we claim that such results are not the effects of the nonlinearity.

To prove this, we solve the problem again using another GS4-1 algorithm defined by $\rho_\infty = 1$ and $\rho_\infty^s = 0$ as the basic primitive algorithm to march the solutions in time. This particular algorithm imposes maximal numerical dissipation on the time derivative variable (by choosing $\rho_\infty^s = 0$), in contrast to the previous nondissipative algorithm (i.e., $\rho_\infty^s = 1$/zero-damping). As we claim that the inconsistent results seen in Figure 6.1 is due to the zero-damping property of the totally nondissipative algorithm, and not due to the nonlinear approaches employed, we suppose that consistent performance of the nonlinear approaches can now be seen with the new basic primitive algorithm defined by $\rho_\infty = 1$ and $\rho_\infty^s = 0$. The results of T and $\dot{T}$ and the corresponding errors generated by the different nonlinear approaches, employing this algorithm as the basic primitive algorithm to march the solution in time, are shown in Figure 6.2. An observation of this figure indicates that the solutions of T generated by this algorithm is more or less the same as those generated by the totally nondissipative algorithm (see Figures 6.2(a) and 6.1(a)). This is to be expected since in both cases, ρ_∞ (which has dominant influence on the primary variable) is the same (i.e., $\rho_\infty = 1$). However, comparison of Figures 6.2(b) and 6.1(b) indicates that, with another GS4-1 algorithm defined by $\rho_\infty = 1$ and $\rho_\infty^s = 0$ used as the basic primitive algorithm to march the solution in time, the classical approach yields the largest error while the normalized time weighted residual approach with the first nonlinear treatment yields the least error. This explains that the oscillatory solutions of $\dot{T}$ and the corresponding errors seen in Figures 6.1(b) and 6.1(d), respectively, are due

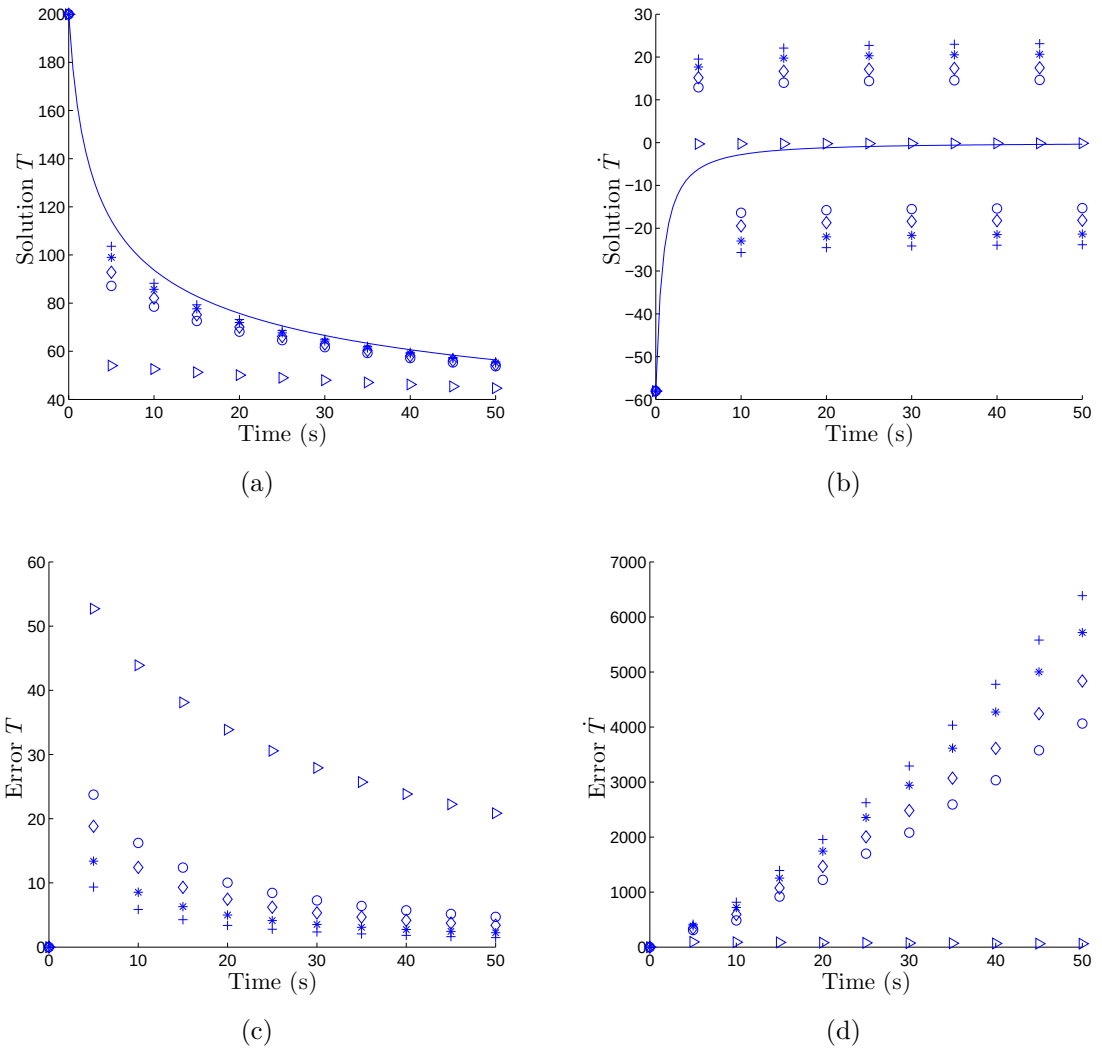


Figure 6.1: Plot of T and $\dot{T}$ (for the radiation heat transfer problem, Section 6.2) and the corresponding errors as a function of time generated with $\Delta t = 5s$ by the nondissipative scheme/GS4-1($\rho_\infty = \rho_\infty^s = 1$) using: $\triangleright$ the classical approach (Section 6.2.2), and the normalized time weighted residual approach (Section 6.2.3) with: $+$ the first nonlinear treatment, $\circ$ the second nonlinear treatment, $\diamond$ the third nonlinear treatment, and $\star$ the fourth nonlinear treatment. $-$ is the exact solution

to the zero-damping property of the totally nondissipative scheme, and not due to the failure of the normalized time weighted residual approach. When the numerical dissipation is imposed on the time derivative variable (by choosing $\rho_\infty^s = 0$ via the selective control feature), we can see from Figure 6.2(b) that the different nonlinear treatments derived from the normalized time weighted residual approach perform well to yield physically representative solutions of $\dot{T}$ and subsequently yield less errors. It is also to be noted that the normalized time weighted residual approach with the first nonlinear treatment consistently yields the best results in comparison to the other nonlinear treatments. From this analysis, we can say that as long as the appropriate basic primitive algorithm is used to march the solutions in time, the normalized time weighted residual approach with the first nonlinear treatment performs the best among all possible approaches, for this particular problem.

To validate the trends and conclusions observed so far, we again solve the problem but using smaller time step size of $\Delta t = 2.5s$ and repeat the above analysis. Results are shown in Figure 6.3 for the solutions generated by the GS4-1 framework with $\rho_\infty = 1$ and $\rho_\infty^s = 0$ (i.e., with the selective control feature). Similar trends are observed, and we again conclude that the normalized time weighted residual approach with the first nonlinear treatment performs the best among all possible approaches, provided that the appropriate basic primitive algorithm is used to march the solution in time. It is also worth mentioning that these different approaches yield the same computational efficiencies, such as the required CPU time and the number of nonlinear iterations required for the solutions to converge. The difference in performance between them is only in terms of the accuracy of the solutions as illustrated above.

Additionally, it is to be noted that the conclusion drawn here is specific for the particular problem considered in this section, which serves as an illustrative example to demonstrate how the GS4-1 framework can be used to solve nonlinear problems, and in particular, how the normalized time weighted residual approach can be employed to properly treat the nonlinear terms involved in the problems. The observations and conclusions may vary for different nonlinear problems. In general, for any nonlinear problems, the analysts have to investigate the following: (1) what

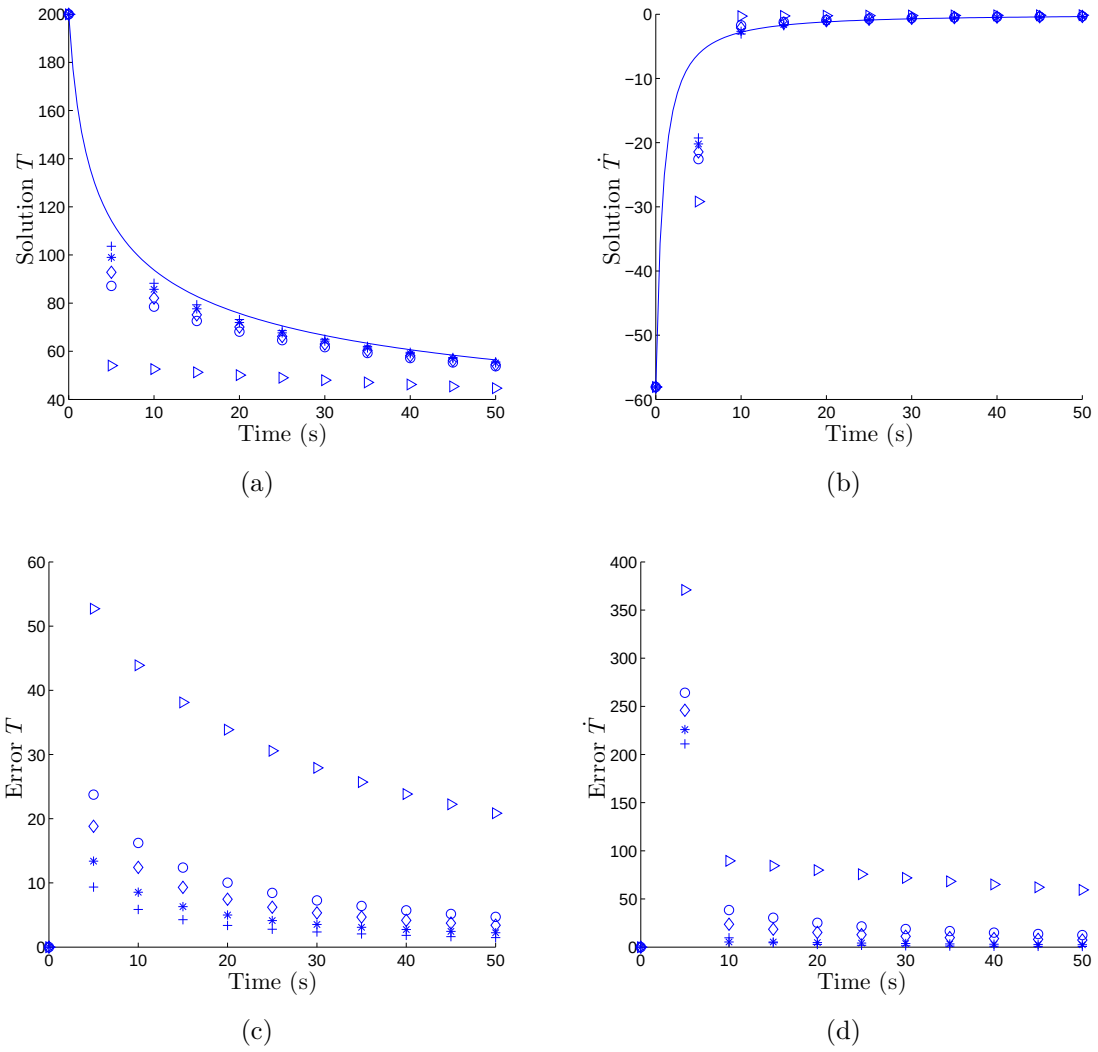


Figure 6.2: Plot of T and $\dot{T}$ (for the radiation heat transfer problem, Section 6.2) and the corresponding errors as a function of time generated with $\Delta t = 5s$ by the GS4-1 framework with $\rho_\infty = 1$ and $\rho^s_\infty = 0$ using: $\triangleright$ the classical approach (Section 6.2.2), and the normalized time weighted residual approach (Section 6.2.3) with: $+$ the first nonlinear treatment, $\circ$ the second nonlinear treatment, $\diamond$ the third nonlinear treatment, and $\star$ the fourth nonlinear treatment. $-$ is the exact solution

the possible different nonlinear treatments are (i.e., of how the nonlinear term can be treated, using the classical and normalized time weighted residual approaches), and subsequently, (2) which nonlinear treatment performs the best among all the possible different approaches for the nonlinear problem of interest. In the absence of analytical solutions, (2) can be done by looking at the resulting conservation properties of quantities such as energy, mass, and momentum [41–44].

Now that we have demonstrated which nonlinear approach yields the best results among all the possible approaches, which is the normalized time weighted residual approach with the first nonlinear treatment described in Section 6.2.3.1, we next employ this approach and proceed to demonstrate the ability and advantage of the newly developed GS4-1 framework via the selective control feature for the present nonlinear application. For this purpose, we solve the problem using the two cases (see *Remark* 4.3.1), i.e., (1) the GS4-1 framework without the selective control feature (i.e., $\rho_\infty = \rho_\infty^s$) and (2) the GS4-1 framework with the selective control feature (i.e., $\rho_\infty \neq \rho_\infty^s$, with $\rho_\infty^s = 0$). As explained previously, we start off by imposing no numerical dissipation on the primary variable, i.e., by choosing $\rho_\infty = 1$. The resulting solutions of T and $\dot{T}$ generated by the two cases are shown in Figure 6.4. From Figure 6.4(a), we can see that both cases yield accurate solutions of T. However, as expected from the above discussion, the former case without the selective control feature results in numerical oscillation in the solution of $\dot{T}$ as seen in Figure 6.4(b). The latter case with the selective control feature, on the other hand, is able to yield physically representative solutions of such a variable via the new feature.

Since the case without the selective control feature requires that $\rho_\infty = \rho_\infty^s$, the only way the method can reduce and eliminate the numerical oscillations in the solutions of $\dot{T}$ is by imposing numerical dissipation on both variables. This can be achieved by choosing $\rho_\infty = \rho_\infty^s < 1$. However, in doing this, caution has to be practised, i.e., not to impose over-dissipation (as discussed before). For purpose of illustration, in Figure 6.5 we show the results generated by this method when the parameters are chosen to be $\rho_\infty = \rho_\infty^s = 0.9$, which means that a slight numerical dissipation is imposed on both the primary variable and its time derivative. For completeness, we also show the results generated by the case with the selective control feature,

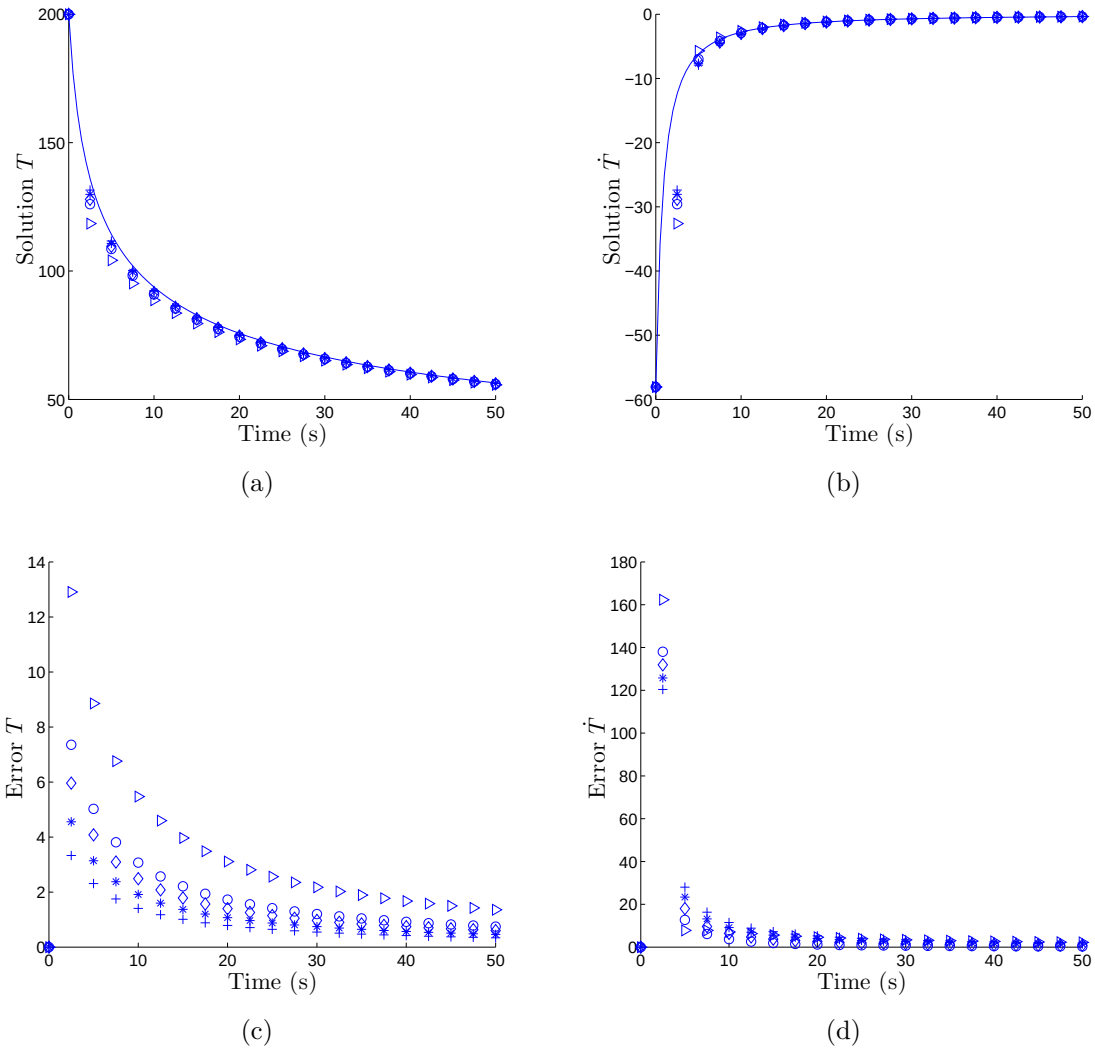
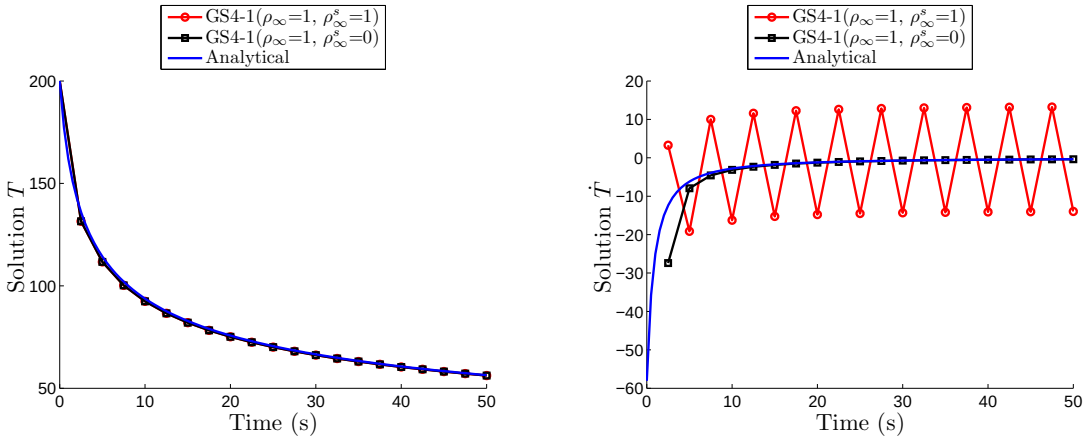


Figure 6.3: Plot of T and $\dot{T}$ (for the radiation heat transfer problem, Section 6.2) and the corresponding errors as a function of time generated with $\Delta t = 2.5s$ by the GS4-1 framework with $\rho_\infty = 1$ and $\rho^s_\infty = 0$ using: $\triangleright$ the classical approach (Section 6.2.2), and the normalized time weighted residual approach (Section 6.2.3) with: $+$ the first nonlinear treatment, $\circ$ the second nonlinear treatment, $\diamond$ the third nonlinear treatment, and $\star$ the fourth nonlinear treatment. $-$ is the exact solution

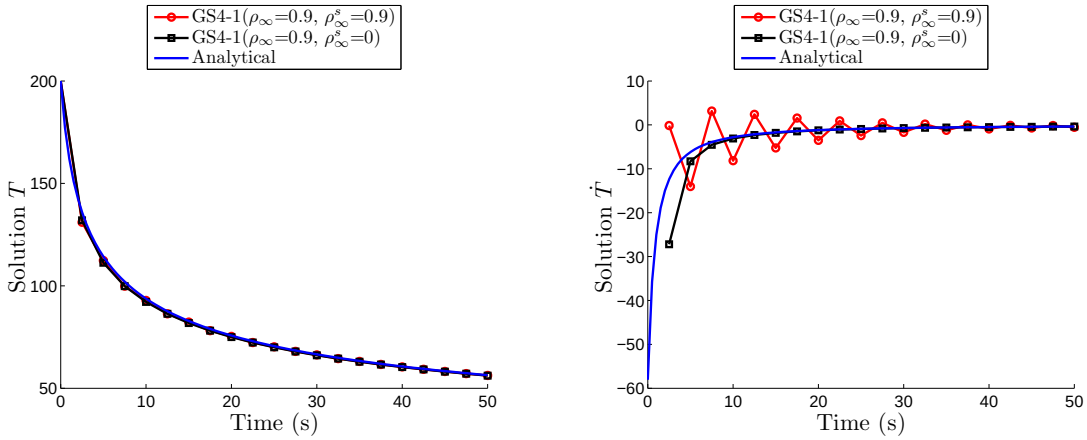


(a) (b)

Figure 6.4: Plot of T and $\dot{T}$ (for the radiation heat transfer problem, Section 6.2) as a function of time generated with $\Delta t = 2.5s$ by: (1) the GS4-1 framework with $\rho^s_\infty = \rho_\infty = 1$ (i.e., the case without the selective control feature), and (2) the GS4-1 framework with $\rho_\infty = 1$ and $\rho^s_\infty = 0$ (i.e., the case with the selective control feature), employing the normalized time weighted residual approach with the first nonlinear treatment (Section 6.2.3.1)

using the same ρ_∞ value and in particular, with $\rho^s_\infty = 0$. Figure 6.5(b) shows that the case without the selective control feature still results in numerical oscillations in $\dot{T}$ even after imposing numerical dissipation. The present development with the new selective control feature, on the other hand, can readily yield physically representative solutions of both T and $\dot{T}$ even without having to impose numerical dissipation on the primary variable, i.e., with $\rho_\infty = 1$ (and $\rho^s_\infty = 0$) as seen earlier in Figure 6.4.

The only way the case without the selective control feature can have $\rho^s_\infty = 0$ is by selecting $\rho_\infty = 0$ as well (since this method requires that $\rho_\infty = \rho^s_\infty$). This means that the algorithm is imposing maximal numerical dissipation on both the primary variable and its time derivative. However, we next demonstrate using this particular example that such a choice may result in imposing over-dissipation on the variables which consequently may yield numerical solution that, although is stable, deviates from the actual dynamics of the problem (in other words, the solution cannot capture the dynamics correctly despite being stable) Figure 6.6 shows the results of



(a) (b)

Figure 6.5: Plot of T and $\dot{T}$ (for the radiation heat transfer problem, Section 6.2) as a function of time generated with $\Delta t = 2.5s$ by: (1) the GS4-1 framework with $\rho^s_\infty = \rho_\infty = 0.9$ (i.e., the case without the selective control feature), and (2) the GS4-1 framework with $rho_\infty = 0.9$ and $\rho^s_\infty = 0$ (i.e., the case with the selective control feature), employing the normalized time weighted residual approach with the first nonlinear treatment (Section 6.2.3.1)

T and $\dot{T}$ generated by the case without the selective control feature when maximal numerical dissipation is imposed (i.e., by choosing $\rho_\infty = \rho^s_\infty = 0$). This figure clearly indicates that, although the solution of $\dot{T}$ is now physically representative, the solution of T on the other hand deviates from the exact solution. Such an issue arises when over-dissipation is imposed on the variable. As seen in Figure 6.4(a) and 6.5(a), the solutions of T generated by this method is already acceptable when no or slight numerical dissipation is imposed on this variable. Therefore, imposing maximal numerical dissipation on this variable (i.e., by selecting $\rho_\infty = 0$) means that the algorithm is over-dissipative, and therefore leads to incorrect dynamics of the problem as seen in Figure 6.6(a). The case with the selective control feature, on the other hand, is able to obtain acceptable results for both T and $\dot{T}$ by choosing $\rho_\infty = 1$ and $\rho^s_\infty = 0$ that is achievable via the selective control feature. This, therefore, demonstrates the ability and advantage of such a new feature inherent in the present developments.

For completeness of the analyses for this nonlinear application, we next explicitly

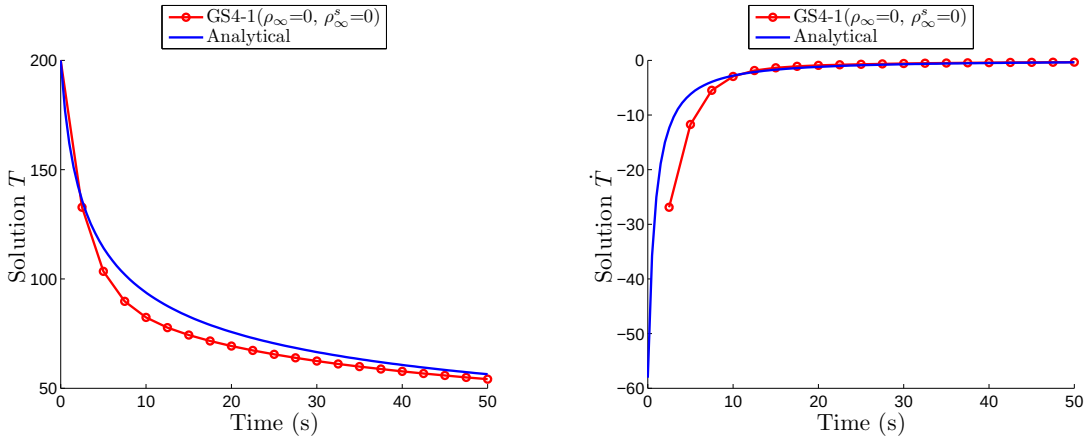


Figure 6.6: Plot of T and $\dot{T}$ (for the radiation heat transfer problem, Section 6.2) as a function of time generated with $\Delta t = 2.5s$ by the GS4-1 framework with $\rho_\infty = \rho^s_\infty = 0$ (i.e., the case without the selective control feature, imposing maximal numerical dissipation), employing the normalized time weighted residual approach with the first nonlinear treatment (Section 6.2.3.1)

demonstrate the rate of convergence in time (i.e., the order of accuracy) of the GS4-1 framework employing the computational procedures described in Section 6.2.4. For this purpose and for consistency, we show the convergence plots of the algorithm that yields the best results for the particular problem considered in this section, namely, GS4-1 framework with $\rho_\infty = 1$ and $\rho^s_\infty = 0$. Additionally, we show results of this algorithm employing the normalized time weighted residual approach with the four different nonlinear treatment as described in Section 6.2.3. The total number of time steps (N_i, see equation (4.5.45)) used to construct the plots are chosen as 100000, 1000, 200, and 50, where the solutions generated with $N_i = 100000$ are used as the reference to calculate the errors for the construction of the convergence plots. Meanwhile, the end time at which the solutions are evaluated is $t_{end} = 50s$. Figure 6.7 shows the convergence plots of T and $\dot{T}$ generated by this GS4-1 algorithm employing the different nonlinear treatments derived from the normalized time weighted residual approach. In contructing these convergence plots, we use the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$, which is the proper construction of the convergence plot as described in Section 4.5.

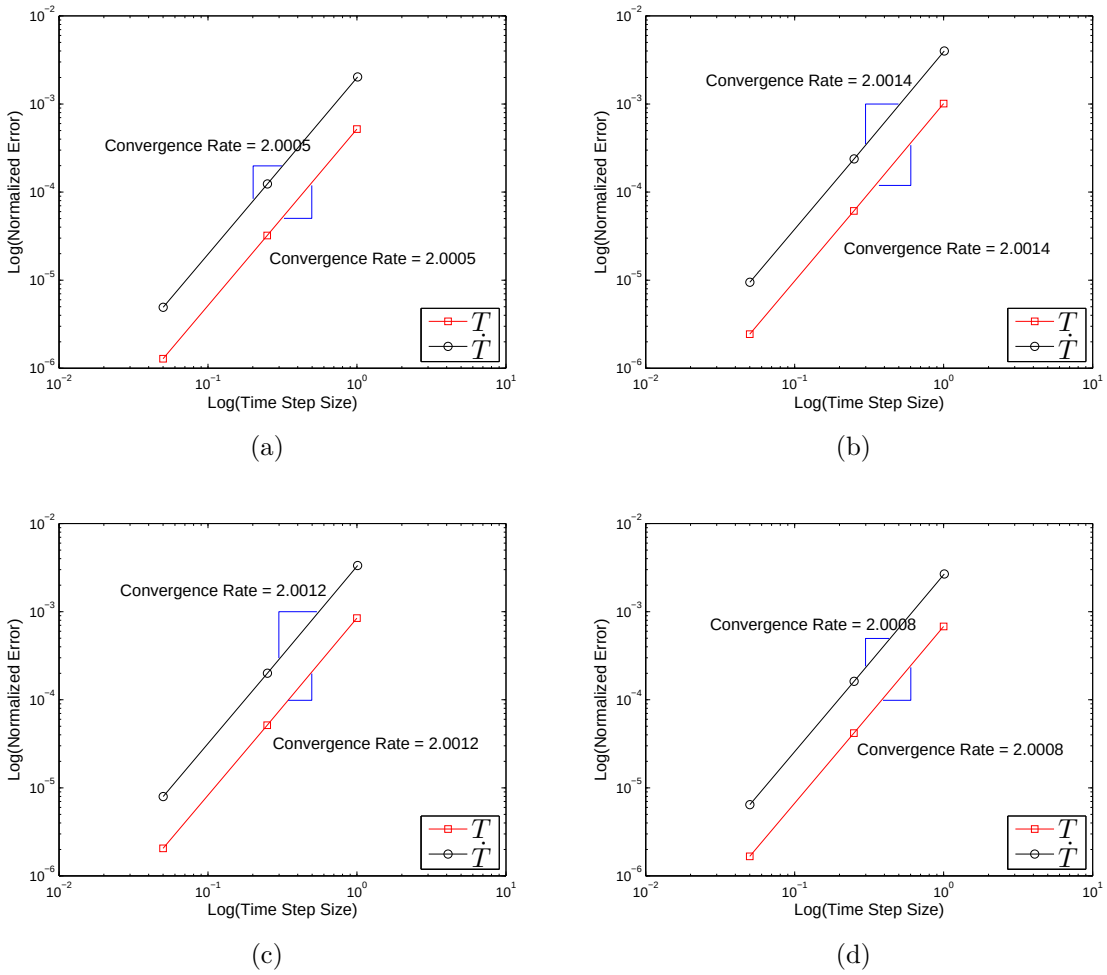


Figure 6.7: Convergence plot (for the radiation heat transfer problem, Section 6.2) of the GS4-1 framework with $\rho_\infty = 1$ and $\rho^s_\infty = 0$ employing the normalized time weighted residual approach with: (a) nonlinear treatment 1 (Section 6.2.3.1), (b) nonlinear treatment 2 (Section 6.2.3.2), (c) nonlinear treatment 3 (Section 6.2.3.3), and (d) nonlinear treatment 4 (Section 6.2.3.4), utilizing the standard convergence plot for T and the time level aligned convergence plot for $\dot{T}$

6.3 Fluid Dynamics

Fluid dynamics is an engineering discipline that deals with the natural science of liquids and gases (so-called fluids) in motion. In this field, most problems are described by the Navier-Stokes equations which are time-dependent in general. Research in this field allows researchers and engineers to handle a wide range of applications, including calculating forces on aircraft, determining the rate of flow of petroleum through pipelines, and predicting the deposition of small particles from liquid suspension onto the pipe wall. Due to its importances, we consider in this section an application of the developed GS4-1 framework to a numerical example that is representative of fluid dynamics applications. This illustrative example serves to provide additional illustrations of how the present GS4-1 framework, originally developed for linear first order transient systems, can be properly extended for use in nonlinear applications with the use of the normalized time weighted residual approach to provide accurate treatment of the nonlinear terms that may appear in this example. Note that the Navier-Stokes equations are very challenging to solve. Therefore, demonstrating the superiority of the new framework for such applications is a significant breakthrough.

6.3.1 Governing Equation

For fluid dynamics type of applications, we consider for purpose of illustration, the application of the GS4-1 framework to the time-dependent Burgers equation, which is well-known to serve as a simplified model of fluid dynamics combining nonlinear advection and linear diffusion. Note that the only missing term in this equation compared to the Navier-Stokes equations is the pressure gradient which is linear. Therefore, the challenge of solving such nonlinear equations remain.[6]

The two dimensional Burgers equation in dimensionless form[7] is given by

[6]Note that since the pressure term does not appear in the Burgers equation, solving such an equation is relatively easier as compared to solving the full Navier-Stokes equations (due to the less number of variables to be solved). For this reason, we consider this simplified model of fluid dynamics in this work. Note that, however, the nonlinearity of the problem remains due to the reason cited above.

[7]note that all variables involved and discussed in the text and figures throughout this section are in dimensionless form

$$\frac{\partial u_i}{\partial t} + \sum_{j=1}^{2} u_j \frac{\partial u_i}{\partial x_j} = \frac{1}{Re} \sum_{j=1}^{2} \frac{\partial^2 u_i}{\partial x_j^2}, \quad \forall x_i \in \Omega \subset R^2 \quad i = 1, 2, \quad t > 0 \tag{6.3.108}$$

where u_i is the i^{th} component of the velocity, Re is the Reynold number, and Ω is a bounded domain in R^2. The physical problem may be specified on the boundary, denoted Γ, by Dirichlet and/or Neumann boundary conditions

$$u_i = u_{i\Gamma} \quad \forall x_i \in \Gamma_1 \quad i = 1, 2, \tag{6.3.109}$$

$$\frac{1}{Re} \sum_{j=1}^{2} \frac{\partial u_i}{\partial x_j} n_j = q_i \quad \forall x_i \in \Gamma_2 \quad i = 1, 2, \tag{6.3.110}$$

$$\Gamma = \Gamma_1 + \Gamma_2 \tag{6.3.111}$$

where n_j is the outer normal on Γ_2 while $u_{i\Gamma}$ and q_i are the known vectors of the prescribed velocities on Γ_1 and flux on Γ_2, respectively. Furthermore, a known initial velocity must be specified to complete the problem description

$$u_i = u_{i0} \quad \forall x_i \in \Omega \quad i = 1, 2, \quad t = 0 \tag{6.3.112}$$

6.3.2 Spatial Discretization by the Finite Element Method

To discretize the described problem in space using the standard Galerkin Finite Element Method, we apply the method of weighted residuals to equation (6.3.108) for an arbitrary element (e) with N nodes bounded by $\Omega^{(e)}$ domain. The velocities of the element are approximated by a linear combination of time-independent element shape functions $\psi_{in}^{(e)} (i = 1, ..., N)$

$$u_i^{(e)} = \sum_{n=1}^{N} u_{in}^{(e)} \psi_{in}^{(e)}, \quad i = 1, 2 \tag{6.3.113}$$

The shape functions $\psi_{in}^{(e)}$ are used as the weighting functions in the weighted residual equation. Employing equation (6.3.113), we obtain the following system of nonlinear

ordinary differential equations in time

$$\mathbf{M}^{(e)}\dot{\mathbf{u}}^{(e)} + \mathbf{P}(\mathbf{u}^{(e)}) + \mathbf{K}^{(e)}\mathbf{u}^{(e)} = \mathbf{F}^{(e)} \tag{6.3.114}$$

where $\mathbf{u}^{(e)}$ a vector of length $2N$ containing the nodal velocities $u_{in}^{(e)}(i = 1, 2; n = 1, ...N)$, while $\dot{\mathbf{u}}^{(e)}$ is the time derivative of $\mathbf{u}^{(e)}$. Furthermore, $\mathbf{M}^{(e)}$ is the element mass matrix defined as ($p = 1, ..., N$ and $q = 1, ..., N$)

$$\mathbf{M}^{(e)} = \begin{bmatrix} \mathbf{M}^{(e)11} & \mathbf{0} \\ \\ \mathbf{0} & \mathbf{M}^{(e)22} \end{bmatrix} \tag{6.3.115}$$

where

$$\mathbf{M}^{(e)ij}(p, q) = \int_{\Omega^{(e)}} \psi_{ip}^{(e)} \psi_{jq}^{(e)} \partial\Omega \tag{6.3.116}$$

Meanwhile, $\mathbf{P}(\mathbf{u}^{(e)})$ is a vector of length $2N$ containing the nonlinear convection term and is defined as

$$\mathbf{P}(\mathbf{u}^{(e)}) = \begin{Bmatrix} \mathbf{P}^{(e)1} \\ \mathbf{P}^{(e)2} \end{Bmatrix} \tag{6.3.117}$$

where ($p = 1, ..., N$ and $i = 1, 2$)

$$\mathbf{P}^{(e)i}(p) = \sum_{j=1}^{2} \int_{\Omega^{(e)}} \left[\sum_{n=1}^{N} u_{jn}^{(e)} \psi_{jn}^{(e)} \sum_{l=1}^{N} u_{il}^{(e)} \frac{\partial \psi_{il}^{(e)}}{\partial x_j} \right] \psi_{ip}^{(e)} \partial\Omega \tag{6.3.118}$$

Note that $\mathbf{K}^{(e)}$ is the diffusion matrix defined as ($p = 1, ..., N$ and $q = 1, ..., N$)

$$\mathbf{K}^{(e)} = \begin{bmatrix} \mathbf{K}^{(e)11} & \mathbf{0} \\ \\ \mathbf{0} & \mathbf{K}^{(e)22} \end{bmatrix} \tag{6.3.119}$$

where

$$\mathbf{K}^{(e)ij}(p,q) = \frac{1}{Re}\int_{\Omega^{(e)}} \sum_{r=1}^{2} \frac{\partial \psi_{ip}^{(e)}}{\partial x_r}\frac{\partial \psi_{jq}^{(e)}}{\partial x_r}\partial\Omega \tag{6.3.120}$$

and $\mathbf{F}^{(e)}$ is the force vector of length $2N$ resulting from integration by parts of the diffusive term in the governing equation. It is defined as

$$\mathbf{F}^{(e)} = \left\{ \begin{array}{c} \mathbf{F}^{(e)1} \\ \mathbf{F}^{(e)2} \end{array} \right\} \tag{6.3.121}$$

where ($p = 1, ..., N$ and $i = 1, 2$)

$$\mathbf{F}^{(e)i}(p) = \int_{\Omega^{(e)}} q_i^{(e)} \psi_{ip}^{(e)} \partial\Omega \tag{6.3.122}$$

Equation (6.3.114) is then assembled for all elements to yield the following global system of first order (in time) ordinary differential equations

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{P}(\mathbf{u}) + \mathbf{K}\mathbf{u} = \mathbf{F} \tag{6.3.123}$$

Equation (6.3.123) is nonlinear due to the presence of the nonlinear term, $\mathbf{P}(\mathbf{u})$ (defined in equation (6.3.118)), arising from the nonlinear convection that appears in the Burgers equation. This nonlinear term requires an accurate treatment in the computational procedure to ensure physically representative and accurate solutions, and in some cases, the convergence of the nonlinear iterations within a time step. As the detailed descriptions of how nonlinear first order transient systems can be discretized in time using both the classical and normalized time weighted residual approaches have been presented in Section 6.2, with particular application to radiation heat transfer problem, in this section, we briefly describe how these approaches can be consistently employed to solve the considered fluid dynamics model to provide additional illustration of the essential concepts. More importantly, our focus in this section is to demonstrate the consistent algorithmic properties of the GS4-1 framework for this nonlinear application as those seen previously.

6.3.3 Classical Time Weighted Residual Approach

Employing the classical time weighted residual approach to equation (6.3.123) in the same manner as done for the linear case, with the GS4-1 framework used as the basic primitive algorithm to march the solutions in time, yields the following temporally discrete equation (for relevant details, please refer to Section 6.2)

$$\mathbf{M}\widetilde{\dot{\mathbf{u}}} + \widetilde{\mathbf{P}} + \mathbf{K}\widetilde{\mathbf{u}} = \widetilde{\mathbf{F}} \tag{6.3.124}$$

where

$$\widetilde{\dot{\mathbf{u}}} = \dot{\mathbf{u}}_n + \Lambda_6 W_1 \Delta\dot{\mathbf{u}} \tag{6.3.125}$$

$$\widetilde{\mathbf{u}} = \mathbf{u}_n + \Lambda_4 W_1 \Delta t \dot{\mathbf{u}}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\mathbf{u}} \tag{6.3.126}$$

$$\widetilde{\mathbf{P}} = \mathbf{P}_n + \Lambda_4 W_1 \Delta t \dot{\mathbf{P}}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\mathbf{P}} \tag{6.3.127}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{6.3.128}$$

The associated expressions for the update of the variables at the end of each time level are chosen as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_4 \dot{\mathbf{u}}_n \Delta t + \lambda_5 \Delta\dot{\mathbf{u}} \Delta t \tag{6.3.129}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \Delta\dot{\mathbf{u}} \tag{6.3.130}$$

$$\mathbf{P}_{n+1} = \mathbf{P}_n + \lambda_4 \dot{\mathbf{P}}_n \Delta t + \lambda_5 \Delta\dot{\mathbf{P}} \Delta t \tag{6.3.131}$$

$$\dot{\mathbf{P}}_{n+1} = \dot{\mathbf{P}}_n + \Delta\dot{\mathbf{P}} \tag{6.3.132}$$

The above formulation is the resulting framework to solve the time-dependent Burgers equation employing the classical time weighted residual approach; and with the GS4-1 framework used as the basic primitive algorithm to march the solutions in time. The algorithmic parameters ($\Lambda_6 W_1$, $\Lambda_5 W_2$, $\Lambda_4 W_1$, W_1, λ_4, and λ_5) can be expressed in terms of ρ_∞ and ρ_∞^s as given by equation (6.2.23). Upon rearrangement, equations (6.3.125) to (6.3.130) can also be represented in another form as follows:

$$\widetilde{\dot{\mathbf{u}}} = \left(1 - \frac{\lambda_4}{\lambda_5}\Lambda_6 W_1\right)\dot{\mathbf{u}}_n + \frac{\Lambda_6 W_1}{\lambda_5}\frac{\Delta \mathbf{u}}{\Delta t} \tag{6.3.133}$$

$$\widetilde{\mathbf{u}} = \mathbf{u}_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta \mathbf{u} \tag{6.3.134}$$

$$\widetilde{\mathbf{P}} = \mathbf{P}_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta \mathbf{P} \tag{6.3.135}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{6.3.136}$$

where

$$\Delta \mathbf{P} = \mathbf{P}_{n+1} - \mathbf{P}_n, \quad \mathbf{P}_n = \mathbf{P}\left(\mathbf{u}_n\right), \quad \mathbf{P}_{n+1} = \mathbf{P}\left(\mathbf{u}_{n+1}\right) \tag{6.3.137}$$

with the corresponding expressions for the updates of the variables at the end of each time level

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta \mathbf{u} \tag{6.3.138}$$

$$\dot{\mathbf{u}}_{n+1} = \frac{1}{\lambda_5}\frac{\Delta \mathbf{u}}{\Delta t} + \left(1 - \frac{\lambda_4}{\lambda_5}\right)\dot{\mathbf{u}}_n \tag{6.3.139}$$

6.3.4 Normalized Time Weighted Residual Approach

We next describe in this section how to employ the normalized time weighted residual approach for the fluid dynamics model considered in this section. Since the nonlinear term involved ($\mathbf{P}$ defined in equation (6.3.118)) is only a square of the primary variable ($\mathbf{u}$), this approach yields only one nonlinear treatment which is described next.

Employing the normalized time weighted residual approach to equation (6.3.123), with the GS4-1 framework used as the basic primitive algorithm to march the solutions in time, yields the following temporally discrete equation

$$\mathbf{M}\widetilde{\dot{\mathbf{u}}} + \mathbf{P}(\widetilde{\mathbf{u}}) + \mathbf{K}\widetilde{\mathbf{u}} = \widetilde{\mathbf{F}} \tag{6.3.140}$$

where

$$\widetilde{\dot{\mathbf{u}}} = \dot{\mathbf{u}}_n + \Lambda_6 W_1 \Delta\dot{\mathbf{u}} \tag{6.3.141}$$

$$\widetilde{\mathbf{u}} = \mathbf{u}_n + \Lambda_4 W_1 \Delta t \dot{\mathbf{u}}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\mathbf{u}} \tag{6.3.142}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{6.3.143}$$

The associated expressions for the update of the variables are consistently chosen as

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_4 \dot{\mathbf{u}}_n \Delta t + \lambda_5 \Delta\dot{\mathbf{u}} \Delta t \tag{6.3.144}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \Delta\dot{\mathbf{u}} \tag{6.3.145}$$

The above formulation is the resulting framework to solve the time-dependent Burgers equation employing the normalized time weighted residual approach and with the GS4-1 framework used as the basic primitive algorithm to march the solutions in time. As done previously for the classical approach, equations (6.3.141) to (6.3.145) can also be represented in another form as follows:

$$\widetilde{\dot{\mathbf{u}}} = \left(1 - \frac{\lambda_4}{\lambda_5}\Lambda_6 W_1\right)\dot{\mathbf{u}}_n + \frac{\Lambda_6 W_1}{\lambda_5}\frac{\Delta\mathbf{u}}{\Delta t} \tag{6.3.146}$$

$$\widetilde{\mathbf{u}} = \mathbf{u}_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta\mathbf{u} \tag{6.3.147}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{6.3.148}$$

with the corresponding update equations

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta\mathbf{u} \tag{6.3.149}$$

$$\dot{\mathbf{u}}_{n+1} = \frac{1}{\lambda_5}\frac{\Delta\mathbf{u}}{\Delta t} + \left(1 - \frac{\lambda_4}{\lambda_5}\right)\dot{\mathbf{u}}_n \tag{6.3.150}$$

6.3.5 Computational Procedures for Both Approaches

The above descriptions for both approaches (i.e., the classical and normalized time weighted residual approaches) are to be completed by the description of an

iterative process such as the Newton-Raphson method to obtain a full description of the computational procedures needed to solve the considered problem. For this problem, we again employ the Newton-Raphson method and present the complete computational procedures in this section. Also, as previously done, we describe these computational procedures in a general form that is common to both approaches described above, and additionally, we mention the differences in the approaches as they arise, in order to improve the clarity of such a description. We now employ the Newton-Raphson method to iteratively solve the equation at each time level. The computational details follow next.

Algorithm 6
Computational Procedures (Both Approaches) for Solving the Burgers Equation, Section 6.3

Given/knowing the solutions at previous time level t_n (i.e., $\mathbf{u}_n$ and $\dot{\mathbf{u}}_n$), we seek the solutions of the nodal primary variable and its time derivative at the next time level t_{n+1} (i.e., $\mathbf{u}_{n+1}$ and $\dot{\mathbf{u}}_{n+1}$). At the beginning of t_{n+1} time level, we initially predict the solutions using known values at previous time level as follows:

$$\begin{aligned} \mathbf{u}_{n+1}^k &= \mathbf{u}_n \\ \dot{\mathbf{u}}_{n+1}^k &= \left(1 - \frac{\lambda_4}{\lambda_5}\right) \dot{\mathbf{u}}_n \end{aligned} \tag{6.3.151}$$

where the superscript k is the nonlinear iteration counter. We then calculate the algorithmic variables $\widetilde{\mathbf{u}}^k$ and $\widetilde{\dot{\mathbf{u}}}^k$ in terms of the predicted values $\mathbf{u}_{n+1}^k$ and $\dot{\mathbf{u}}_{n+1}^k$ using equations (6.3.134) *and* (6.3.125)*, respectively, as follows*

$$\widetilde{\mathbf{u}}^k = \mathbf{u}_n + \frac{\Lambda_5 W_2}{\lambda_5}(\mathbf{u}_{n+1}^k - \mathbf{u}_n) \tag{6.3.152}$$

$$\widetilde{\dot{\mathbf{u}}}^k = \dot{\mathbf{u}}_n + \Lambda_6 W_1(\dot{\mathbf{u}}_{n+1}^k - \dot{\mathbf{u}}_n) \tag{6.3.153}$$

while the algorithmic load term $\widetilde{F}^k$ *is determined from its known values at* t_n *and* t_{n+1} *as follows*

$$\widetilde{F}^k = F_n + W_1(F_{n+1} - F_n) \tag{6.3.154}$$

We next enter the nonlinear iteration loop. This is done as follows:

(i) In this nonlinear iteration loop, we first calculate the residual resulting from using the predicted algorithmic variables at the k-th iteration. The residual at the k-th iteration $\widetilde{\mathbf{R}}^k$ *can be represented in a general form that is common to all approaches as follows*

$$\widetilde{\mathbf{R}}^k = \mathbf{M}\widetilde{\dot{\mathbf{u}}}^k + \widetilde{\mathbf{P}}^k + \mathbf{K}\widetilde{\mathbf{u}}^k - \widetilde{\mathbf{F}}^k \tag{6.3.155}$$

where $\widetilde{\mathbf{P}}^k$ *represents the algorithmic nonlinear term (in general form) at the k-th iteration. However, as previously mentioned and described in the preceeding section, the two different nonlinear approaches have different definitions of this term due to the different ways this term is represented in these approaches. This is summarized as follows:*

- *For the classical time weighted residual approach (Section 6.3.3), the algorithmic nonlinear term* $(\widetilde{\mathbf{P}}^k)$ *is given by*

$$\widetilde{\mathbf{P}}^k = \mathbf{P}_n + \frac{\Lambda_5 W_2}{\lambda_5}\Delta\mathbf{P} \tag{6.3.156}$$

  where

$$\Delta\mathbf{P} = \mathbf{P}^k_{n+1} - \mathbf{P}_n, \quad \mathbf{P}_n = \mathbf{P}\left(\mathbf{u}_n\right), \quad \mathbf{P}^k_{n+1} = \mathbf{P}\left(\mathbf{u}^k_{n+1}\right) \tag{6.3.157}$$

- *For the normalized time weighted residual approach (Section 6.3.4), this term is given by*

$$\widetilde{\mathbf{P}}^k = \mathbf{P}(\widetilde{\mathbf{u}}^k) \tag{6.3.158}$$

(ii) We next linearize the residual using Taylor series expansion, truncate after the linear term and set the residual to vanish. This step yields the following equation

$$\frac{\partial \widetilde{\mathbf{R}}^k}{\partial \widetilde{\mathbf{u}}^k}\Delta\widetilde{\mathbf{u}} = -\widetilde{\mathbf{R}}^k \tag{6.3.159}$$

where $\Delta\widetilde{\mathbf{u}} = \widetilde{\mathbf{u}}^{k+1} - \widetilde{\mathbf{u}}^k$, *and* $\frac{\partial \widetilde{\mathbf{R}}^k}{\partial \widetilde{\mathbf{u}}^k}$ *is the Jacobian given by*

$$\begin{aligned}\frac{\partial \widetilde{\mathbf{R}}^k}{\partial \widetilde{\mathbf{u}}^k} &= \mathbf{M}\frac{\partial \widetilde{\dot{\mathbf{u}}}^k}{\partial \widetilde{\mathbf{u}}^k} + \frac{\partial \widetilde{\mathbf{P}}^k}{\partial \widetilde{\mathbf{u}}^k} + \mathbf{K}\frac{\partial \widetilde{\mathbf{u}}^k}{\partial \widetilde{\mathbf{u}}^k} - \frac{\partial \widetilde{\mathbf{F}}^k}{\partial \widetilde{\mathbf{u}}^k} \\ &= \mathbf{M}\left(\frac{\Lambda_6 W_1}{\Lambda_5 W_2 \Delta t}\right) + \frac{\partial \widetilde{\mathbf{P}}^k}{\partial \widetilde{\mathbf{u}}^k} + \mathbf{K}\end{aligned} \tag{6.3.160}$$

The term $\frac{\partial \widetilde{\mathbf{P}}^k}{\partial \widetilde{\mathbf{u}}^k}$ *in equation* (6.3.160) *is the tangential stiffness. Since* $\widetilde{\mathbf{P}}^k$ *is different for the two different nonlinear approaches as given by equations* (6.3.156) *and* (6.3.158), *respectively, the corresponding tangential stiffness for each approach is also different. This is summarized as follows*

- *For the classical approach, the tangential stiffness (*$\frac{\partial \widetilde{\mathbf{P}}^k}{\partial \widetilde{\mathbf{u}}^k}$*) can be expressed from the use of equation* (6.3.156) *as follows*

$$\frac{\partial \widetilde{\mathbf{P}}^k}{\partial \widetilde{\mathbf{u}}^k} = \frac{\partial \mathbf{P}(\mathbf{u}^k_{n+1})}{\partial \mathbf{u}^k_{n+1}} \tag{6.3.161}$$

  Note that the term $\frac{\partial \mathbf{P}(\mathbf{u}^k_{n+1})}{\partial \mathbf{u}^k_{n+1}}$ *in equation* (6.3.161) *is the derivative of the nonlinear vector* $\mathbf{P}$*(equation* (6.3.118) *in terms of the predicted value of the primary variable for* t_{n+1} *time level,* $\mathbf{u}^k_{n+1}$*) with respect to* $\mathbf{u}^k_{n+1}$. *Because the evaluation of* $\mathbf{P}$ *depends on the choice of the element shape function (see equation* (6.3.118)*), a closed form expression for this term will also depend on this choice; and therefore will be given in Section 6.3.6 where we discuss the benchmark numerical examples governed by the time-dependent Burgers equation that is described in this section.*

- *Meanwhile, for the normalized time weighted residual approach, the tangential*

stiffness can be expressed as (from use of equation (6.3.158)*)*

$$\frac{\partial \widetilde{\mathbf{P}}^k}{\partial \widetilde{\mathbf{u}}^k} = \frac{\partial \mathbf{P}(\widetilde{\mathbf{u}}^k)}{\partial \widetilde{\mathbf{u}}^k} \tag{6.3.162}$$

Here, the term $\frac{\partial \mathbf{P}(\widetilde{\mathbf{u}}^k)}{\partial \widetilde{\mathbf{u}}^k}$ *is the derivative of the nonlinear vector* $\mathbf{P}$*(equation* (6.3.118) *in terms of the algorithmic primary variable,* $\widetilde{\mathbf{u}}^k$*) with respect to* $\widetilde{\mathbf{u}}^k$*. A closed form expression for this term will also be given in Section 6.3.6 due to the reason cited above.*

Equation (6.3.159) *can be rearranged to the following form*

$$\widetilde{\mathbf{u}}^{k+1} = \left(\frac{\partial \widetilde{\mathbf{R}}^k}{\partial \widetilde{\mathbf{u}}^k}\right)^{-1} \left(-\widetilde{\mathbf{R}}^k + \frac{\partial \widetilde{\mathbf{R}}^k}{\partial \widetilde{\mathbf{u}}^k}\widetilde{\mathbf{u}}^k\right) \tag{6.3.163}$$

(iii) We next impose the Dirichlet boundary condition (if any) to equation (6.3.163)*, and this can be done from the use of equation* (6.3.152)*, i.e., by imposing the* $\widetilde{\mathbf{u}}^{k+1}$ *of the prescribed nodes to take the following value*

$$\widetilde{\mathbf{u}}^{k+1} = \mathbf{u}_n + \frac{\Lambda_5 W_2}{\lambda_5}(\mathbf{u}_{\Gamma n+1} - \mathbf{u}_n) \tag{6.3.164}$$

where $\mathbf{u}_{\Gamma n+1}$ *is the known (prescribed) velocity at the boundary* Γ *(see equation* (6.3.109)*) at time* t_{n+1}*.*

(iv) We then solve for $\widetilde{\mathbf{u}}^{k+1}$ *from equation* (6.3.163) *after imposing the boundary conditions appropriately.*

(v) Subsequently, we correct the algorithmic time derivative variable and the predicted value of the primary variable using equations (6.3.133) *and* (6.3.134)*, respectively, as follows:*

$$\widetilde{\dot{\mathbf{u}}}^{k+1} = \left(1 - \frac{\lambda_4}{\lambda_5}\Lambda_6 W_1\right)\dot{\mathbf{u}}_n + \frac{\Lambda_6 W_1}{\Lambda_5 W_2}\frac{(\widetilde{\mathbf{u}}^{k+1} - \mathbf{u}_n)}{\Delta t} \tag{6.3.165}$$

$$\mathbf{u}_{n+1}^{k+1} = \mathbf{u}_n + \frac{\lambda_5}{\Lambda_5 W_2}(\widetilde{\mathbf{u}}^{k+1} - \mathbf{u}_n) \tag{6.3.166}$$

(vi) Upon obtaining the above algorithmic variables, we then check if convergence of the solution is achieved

$$|\widetilde{\mathbf{u}}^{k+1} - \widetilde{\mathbf{u}}^{k}| = tol \tag{6.3.167}$$

where $\widetilde{\mathbf{u}}^k$ is the value of the algorithmic primary variable at the previous nonlinear iteration, and tol is the user-specified tolerance value (i.e., convergence criterion for the Newton-Raphson iteration).

We repeat the nonlinear iteration (i.e., steps (i) to (vi) described above) until the solutions are converged (i.e., until equation (6.3.167) *is satisfied). Upon convergence, we update the variables at the end of t_{n+1} time level as follows*[8]

$$\begin{aligned}
\mathbf{u}_{n+1} &= \mathbf{u}_n + \frac{\lambda_5}{\Lambda_5 W_2}(\widetilde{\mathbf{u}}^{k+1} - \mathbf{u}_n) \\
\dot{\mathbf{u}}_{n+1} &= \frac{(\mathbf{u}_{n+1} - \mathbf{u}_n)}{\lambda_5 \Delta t} + \left(1 - \frac{\lambda_4}{\lambda_5}\right) \dot{\mathbf{u}}_n
\end{aligned} \tag{6.3.168}$$

Remark 6.3.1

In the computational procedures presented above, it is worthy to note that the two nonlinear approaches (i.e., the classical approach described in Section 6.3.3, and the normalized time weighted residual approach described in Section 6.3.4) have a common computational procedure to solve the nonlinear problem of interest using the GS4-1 framework as the basic primitive algorithm to march the solutions in time, and using the Newton-Raphson method to iteratively arrive at the solutions at each time level. However, it is to be noted that the two approaches treat the nonlinear terms differently as seen in equations (6.3.156) *and* (6.3.158)*, respectively. Subsequently, the two approaches also have different definitions of the tangential stiffness involved in the computational procedure as seen in equations* (6.3.161) *and* (6.3.162)*,*

[8] note that the latter is taken from equation (6.3.150)

respectively.

6.3.6 Numerical Illustrations

In this section, we illustrate the significance of the selective control feature (i.e., by allowing $\rho_\infty \neq \rho_\infty^s$) inherent in the GS4-1 framework as compared to the case without such a feature (i.e., $\rho_\infty = \rho_\infty^s$) by solving two benchmark numerical examples governed by the transient Burgers equation which is representative of the fluid dynamics type of applications. This is described next.

6.3.6.1 1D Burgers Equation

The first example is the one-dimensional transient Burgers equation described in [78]. We intentionally simplify the problem by choosing small Re number to eliminate any need for upwinding in the spatial discretization procedure.

The problem is governed by equation (6.3.108) (in one-dimensional) and has the following initial and boundary conditions (in dimensionless form)

$$u(x,0) = \frac{1}{1+\exp\left(\frac{Re}{4}(2x)\right)} \tag{6.3.169}$$

$$u(0,t) = \frac{1}{1+\exp\left(\frac{Re}{4}(-t)\right)} \tag{6.3.170}$$

$$u(1,t) = \frac{1}{1+\exp\left(\frac{Re}{4}(2-t)\right)} \tag{6.3.171}$$

where the exact solution is given by [78] (in dimensionless form)

$$u(x,t) = \frac{1}{1+\exp\left(\frac{Re}{4}(2x-t)\right)} \tag{6.3.172}$$

For this problem, we choose $Re = 0.1$, and we use 1D linear elements whose element shape function is given by

$$\psi^{(e)} = \begin{bmatrix} 1 - \dfrac{x}{l} & \dfrac{x}{l} \end{bmatrix} \tag{6.3.173}$$

where l is the length of each element. We discretize the spatial domain using 30 elements such that the Galerkin FEM can be appropriately used. This choice of element shape function results in the following elemental nonlinear vector ($\mathbf{P}$, defined in equation (6.3.118)) for this particular problem

$$\mathbf{P}^{(e)} = \frac{1}{6} \begin{Bmatrix} -2u_1^2 + u_1 u_2 + u_2^2 \\ -u_1^2 - u_1 u_2 + 2u_2^2 \end{Bmatrix} \tag{6.3.174}$$

where u_1 and u_2 are the values of the nodal primary variable at nodes 1 and 2, respectively, for each element. Using equation (6.3.174), we can find the derivative of this vector with respect to the primary variable (in a general form) as follows

$$\frac{\partial \mathbf{P}(\mathbf{u})}{\partial \mathbf{u}}^{(e)} = \frac{1}{6} \begin{bmatrix} -4u_1 + u_2 & u_1 + 2u_2 \\ \\ -2u_1 - u_2 & -u_1 + 4u_2 \end{bmatrix} \tag{6.3.175}$$

From the use of equation (6.3.175), therefore, we have the following for the computation of the Jacobian in the Newton-Raphson iteration for the classical approach (see equation (6.3.161))

$$\frac{\partial \mathbf{P}(\mathbf{u}_{n+1}^k)}{\partial \mathbf{u}_{n+1}^k}^{(e)} = \frac{1}{6} \begin{bmatrix} -4(u_1)_{n+1}^k + (u_2)_{n+1}^k & (u_1)_{n+1}^k + 2(u_2)_{n+1}^k \\ \\ -2(u_1)_{n+1}^k - (u_2)_{n+1}^k & -(u_1)_{n+1}^k + 4(u_2)_{n+1}^k \end{bmatrix} \tag{6.3.176}$$

where $(u_1)_{n+1}^k$ and $(u_2)_{n+1}^k$ are the predicted values of the nodal primary variable at nodes 1 and 2, respectively, for each element. Meanwhile, for the normalized time weighted residual approach, we have

$$\frac{\partial \mathbf{P}(\widetilde{\mathbf{u}}^k)}{\partial \widetilde{\mathbf{u}}^k}^{(e)} = \frac{1}{6} \begin{bmatrix} -4\widetilde{u}_1^k + \widetilde{u}_2^k & \widetilde{u}_1^k + 2\widetilde{u}_2^k \\ \\ -2\widetilde{u}_1^k - \widetilde{u}_2^k & -\widetilde{u}_1^k + 4\widetilde{u}_2^k \end{bmatrix} \tag{6.3.177}$$

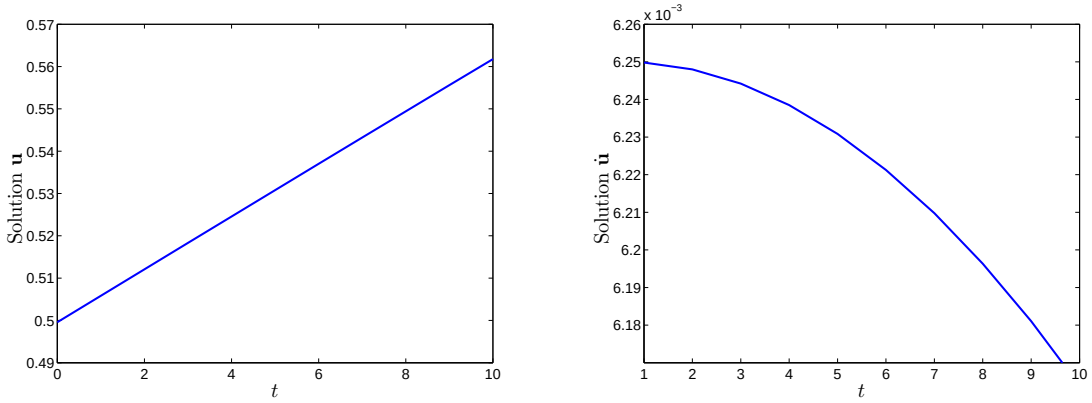
where $\widetilde{u}_1^k$ and $\widetilde{u}_2^k$ are the nodal values of the algorithmic primary variable at nodes 1 and 2, respectively, for each element.

For this problem, we first demonstrate the effect of the selective control feature in the GS4-1 framework. For this purpose, we solve the problem using the two cases (i.e., the GS4-1 framework with and without the selective control feature) defined in *Remark* 4.3.1 as the basic primitive algorithm to march the solutions in time, with a time step size of $\Delta t = 1$ and an end time of $t = 10$. Meanwhile, the convergence criterion (tolerance) for the Newton-Raphson iteration is $tol = 1 \times 10^{-12}$. For the purpose of this investigation, we choose to employ the normalized time weighted residual approach described in Section 6.3.4. The performance of the two cases is first evaluated by comparing the solutions of u and $\dot{u}$ generated by these two cases as a function of time for a specific node at $x = 0.2667$. We first look at the numerical solutions when no numerical dissipation is imposed on the primary variable, i.e., with $\rho_\infty = 1$. Figure 6.8 illustrates the analytical solutions of these variables as a function of time for this particular node as given by equation (6.3.172). Meanwhile, Figure 6.9 shows the numerical solutions generated by the two algorithms. It is clear from Figures 6.9(a) and 6.9(c) that the numerical solutions of the primary variable (u) generated by both cases agree well with the analytical solution even when no numerical dissipation is imposed on the primary variable (i.e., $\rho_\infty = 1$). This is to be expected since the illustrative problem considered here does not involve any shock/sudden change at the boundaries. Instead, the initial and boundary conditions are both derived directly from the analytical solution (see equation (6.3.169) to (6.3.172)). However, Figure 6.9(d) indicates that the numerical solution of the time derivative variable ($\dot{u}$) generated by the case without the selective control feature (i.e., $\rho_\infty = \rho_\infty^s = 1$ in this case) results in large oscillations[9]. On the other hand, Figure 6.9(b) shows that the GS4-1 framework with the selective control feature (i.e., $\rho_\infty = 1, \rho_\infty^s = 0$ in this case) is able to yield good agreement with the analytical solution[10] with the same ρ_∞ value. In other words, the GS4-1 framework with the selective control feature is able to successfully suppress the numerical oscillations in

[9]it is to be noted that due to the automatic scale generated by the plotting routine employed, although not representative, this oscillation makes the analytical solution simply appears as a straight line due to the different ranges of the $\dot{u}$ solution values; see Figure 6.9(d)

[10]hence is capable to capture the analytical solution curve; see Figure 6.9(b)

both u and $\dot{u}$ hence is capable of capturing the dynamics of the problem correctly via such an important and useful feature that is newly introduced in the present development.



(a) (b)

Figure 6.8: Plot of analytical solutions u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) as a function of time for a specific node at $x = 0.2667$ as given by equation (6.3.172)

Since the method without the selective control feature (i.e., $\rho_\infty = \rho_\infty^s$) yields numerical oscillation in the time derivative variable when no numerical dissipation is imposed (i.e., $\rho_\infty = \rho_\infty^s = 1$), the only way the method can reduce and subsequently eliminate this oscillation is by imposing numerical dissipation on this variable (i.e., by choosing $\rho_\infty^s < 1$). However, imposing numerical dissipation on the time derivative variable also means that the same amount of numerical dissipation is imposed on the primary variable due to the limited control feature of such a method, which requires that $\rho_\infty = \rho_\infty^s$. As discussed before, in doing this, precaution has to be practised by imposing only minimal amount of numerical dissipation that is required to eliminate the numerical oscillation. This is to avoid imposing over-dissipation to these variables as it may yield numerical solutions that incorrectly represent the dynamics of the systems as demonstrated previously in Section 6.2. We show, for illustration purpose only, the solutions generated by the method when a slight numerical dissipation is imposed by choosing $\rho_\infty = \rho_\infty^s = 0.9$ with the same time step size and end time, and for completeness, we also show solutions generated by the present development (i.e., the GS4-1 framework with $\rho_\infty = 0.9$ and $\rho_\infty^s = 0$). The results are shown in

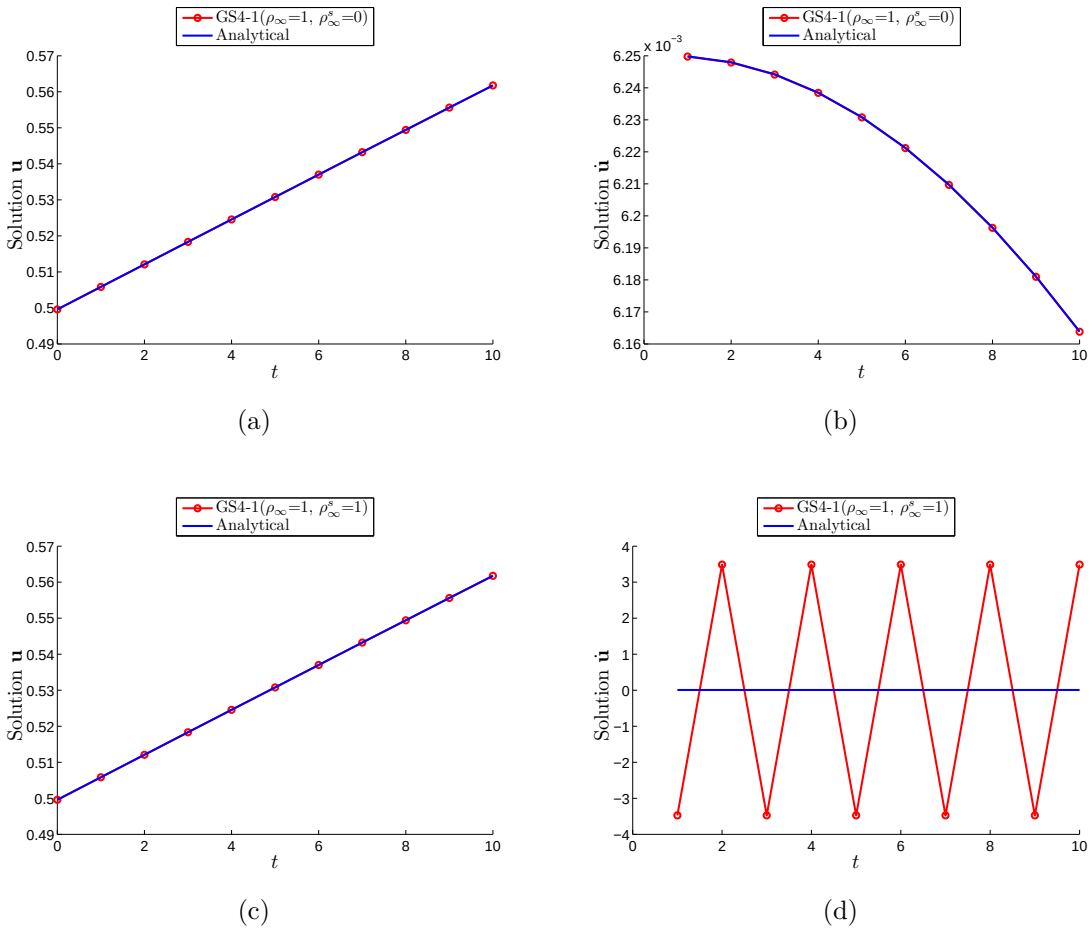


Figure 6.9: Plot of u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) as a function of time for a specific node at $x = 0.2667$ and $\Delta t = 1$ generated by: (i) GS4-1($\rho_\infty = 1, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 1, \rho^s_\infty = 1$) i.e., the case without the selective control feature, employing the normalized time weighted residual method (Section 6.3.4)

Figure 6.10. From this figure, we can see that the solutions of u generated by the two cases agree well with the analytical solution (see Figures 6.10(a) and 6.10(c)). Focusing on the numerical solution of $\dot{u}$, we can see that the solutions generated by the case with the selective control feature is acceptable (see Figure 6.10(b)), while the solution generated by the case without the selective control feature is still oscillating (see Figure 6.10(d)) although the amount of oscillation is slightly reduced (as compared to Figure 6.9(d)) due to the effect of imposing numerical dissipation into the algorithm.

Further attempt is made to investigate the performance of the algorithm without the selective control feature (i.e., GS4-1 framework with $\rho_\infty = \rho_\infty^s$) when the numerical dissipation is further increased to approach the maximal value (i.e. $\rho_\infty \to 0$). For this purpose, we solve the problem using $\rho_\infty = 0.1$ with the same time step size and show results of u and $\dot{u}$ generated by the two cases in Figure 6.11. As expected, the numerical solution of u generated by the two cases consistently agree well with the analytical solution as seen in Figure 6.11(a) and 6.11(c). However, Figure 6.11(d) shows that the algorithm without the selective control feature (i.e. $\rho_\infty = \rho_\infty^s = 0.1$ in this case) still yields oscillation in the numerical solution of $\dot{u}$ even with almost maximal numerical dissipation (i.e., $\rho_\infty = \rho_\infty^s = 0.1$). It is only when this method strictly imposes the maximal numerical dissipation on both u and $\dot{u}$ (i.e., $\rho_\infty = \rho_\infty^s = 0$) that it can eliminate such an oscillation. However, imposing maximal numerical dissipation on the primary variable u means imposing over-dissipation, since the numerical solution of this variable can already/easily be obtained with good accuracy without having to impose any numerical dissipation (i.e., with choice of $\rho_\infty = 1$, see Figures 6.9(a) and 6.9(c)). In this regard, caution has to be exercised since numerical dissipation is only an artifact; while it is handy in one way if one could control it smartly, in another way over-dissipation may also lead to obtaining numerical results that incorrectly represent the dynamics of the system as demonstrated before. Such an issue does not rise in the present GS4-1 framework with the new selective control feature, since the numerical dissipation is selectively imposed, as deemed necessary, on the primary variable and its time derivative, respectively, to yield physically representative solutions of these variables without imposing over-dissipation. For this particular problem, the present approach

is capable of yielding accurate solutions of both the primary variable and its time derivative, even without having to impose numerical dissipation on the primary variable (i.e., $\rho_\infty = 1$). This can easily be achieved by the newly developed GS4-1 framework via the new selective control feature, hence demonstrating the clear improvement and advantage over the current state-of-the-art method (without such a feature).

We next demonstrate that the numerical solutions generated by the GS4-1 framework with the selective control feature, in particular with $\rho_\infty = 1$ and $\rho_\infty^s = 0$, indeed are physically accurate. For this purpose, we show the solutions of both u and $\dot{u}$ for the whole spatial domain at a specific time of $t = 10$ and using a time step size of $\Delta t = 1$. Figure 6.12 shows the analytical solutions of these variables as a function of x at this specific time as given by equation (6.3.172). For comparison purpose, we also show the numerical solutions of these variables as generated by the method without the selective control feature. The results are illustrated in Figure 6.13, from which a comparison of the performance between these two different algorithmic structures can be made. We can see from this figure that the present approach with the selective control feature does yield numerical results that correctly represent the actual dynamics of the problem (see Figures 6.13(a) and 6.13(b)). On the other hand, the case without the selective control feature results in oscillation in the numerical solution of the time derivative variable $\dot{u}$ (see Figure 6.13(d)).[11]

Having demonstrated the effect of the selective control feature for this problem, we next investigate the performance of the different nonlinear treatments (i.e., the classical and normalized time weighted residual approaches described in Sections 6.3.3 and 6.3.4, respectively). For this purpose, we use the GS4-1 framework with the selective control feature (i.e., with particular choice of $\rho_\infty^s = 0$ while ρ_∞ is chosen for illustration purpose as 1, 0.9, 0.8, 0.7, 0.6, and 0.5) as the basic primitive algorithm to march the solutions in time. We solve the problem using a time step size of $\Delta t = 1$ and an end time of $t = 10$. For comparison purpose, we look at the errors in u and $\dot{u}$ generated by both nonlinear treatments for the whole spatial domain at the end time

[11] it is to be noted again that the oscillation generated by this case makes the analytical solution simply appears as a straight line due to the different ranges of the $\dot{u}$ solution values

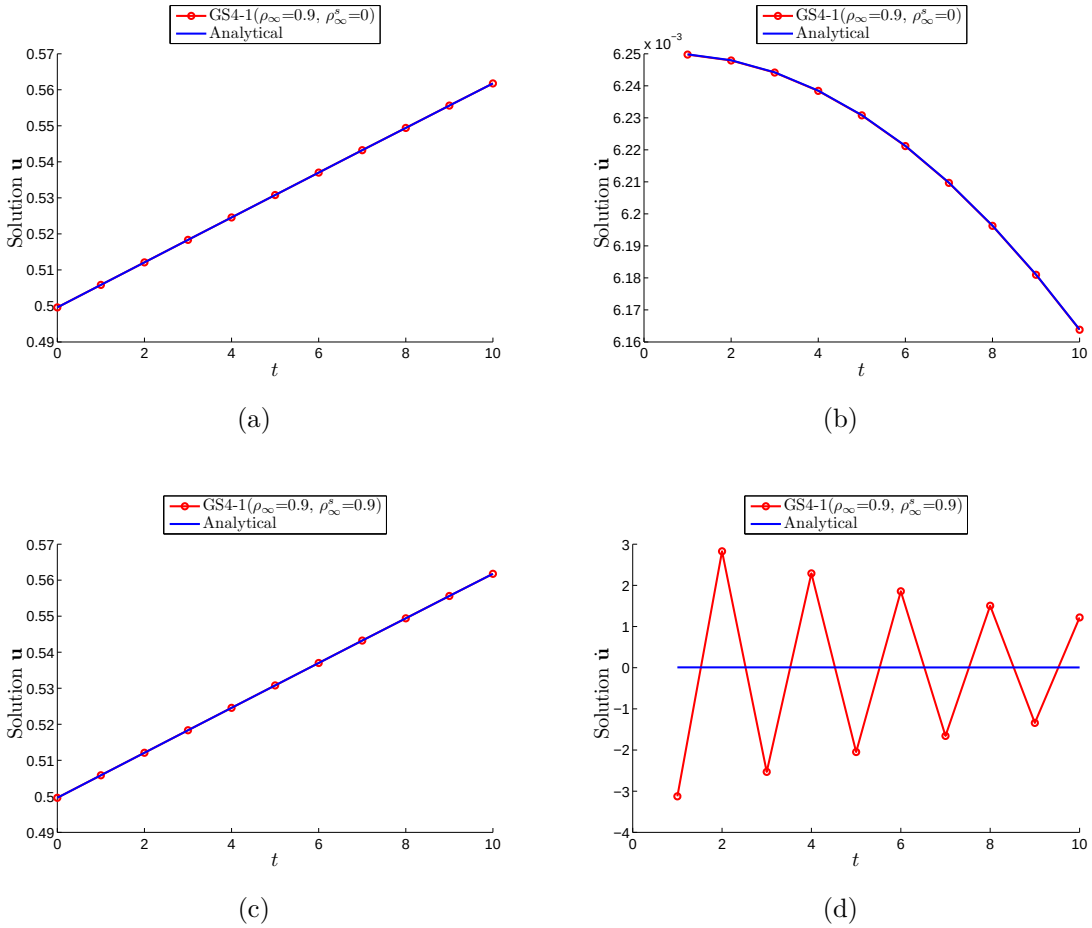


Figure 6.10: Plot of u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) as a function of time for a specific node at $x = 0.2667$ and $\Delta t = 1$ generated by: (i) GS4-1($\rho_\infty = 0.9, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 0.9, \rho^s_\infty = 0.9$) i.e., the case without the selective control feature, employing the normalized time weighted residual method (Section 6.3.4)

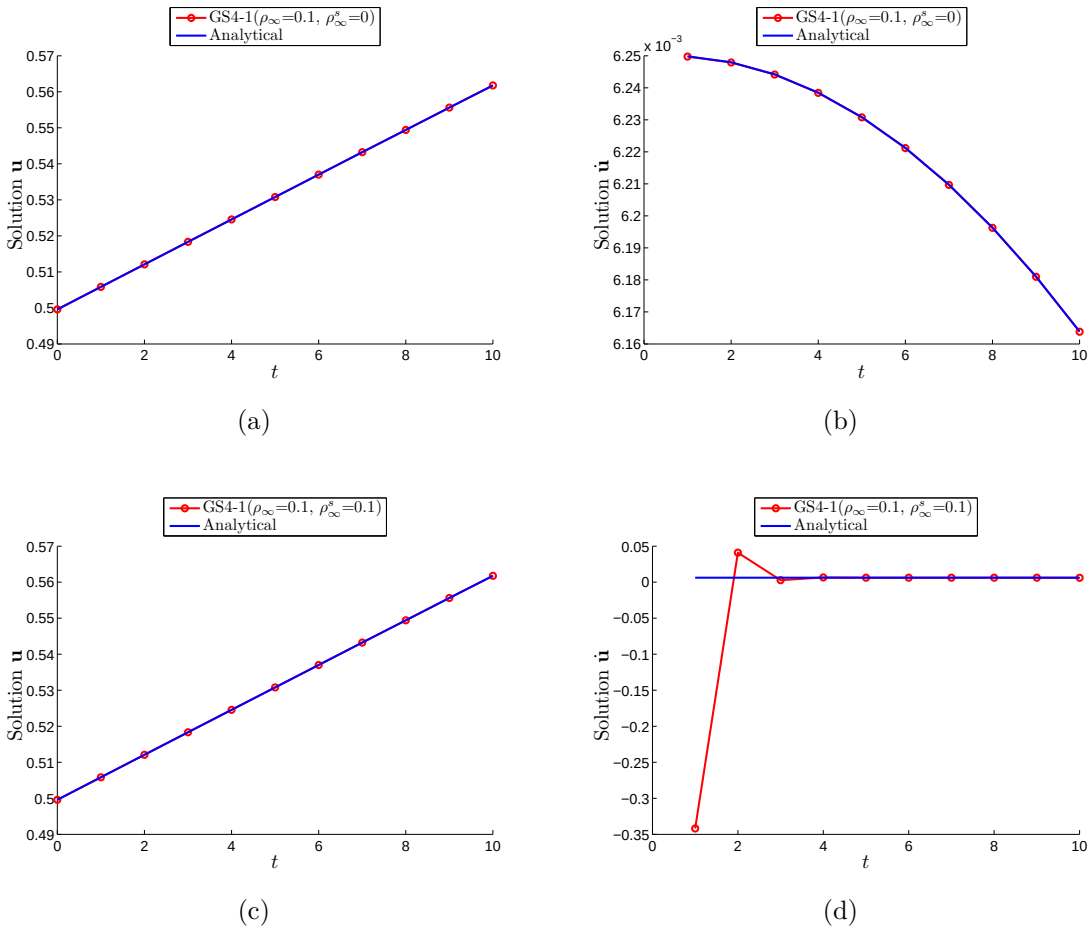
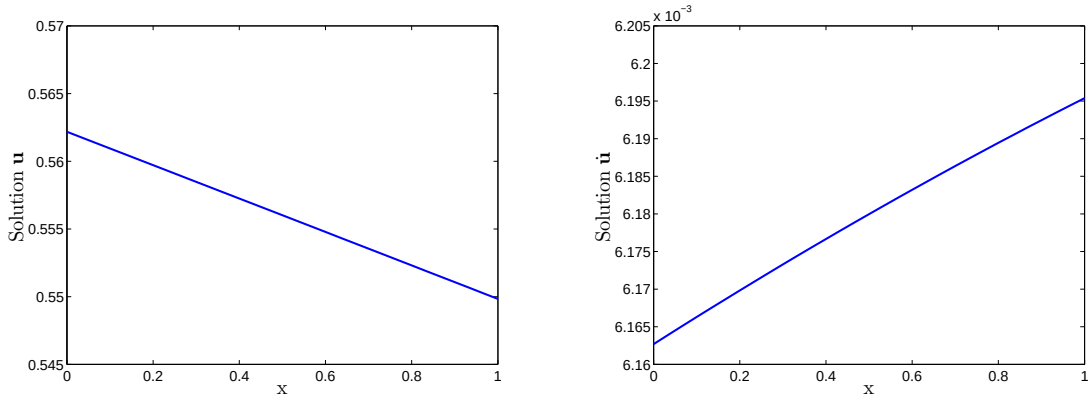


Figure 6.11: Plot of u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) as a function of time for a specific node at $x = 0.2667$ and $\Delta t = 1$ generated by: (i) GS4-1($\rho_\infty = 0.1, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 0.1, \rho^s_\infty = 0.1$) i.e., the case without the selective control feature, employing the normalized time weighted residual method (Section 6.3.4)



(a) (b)

Figure 6.12: Plot of analytical solutions u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) as a function of x for a specific time of $t = 10$ as given by equation (6.3.172)

($t = 10$). The error is defined as

$$Error = \left| \frac{Numerical - Exact}{Exact} \right| \tag{6.3.178}$$

The errors generated by the two nonlinear treatments are shown in Figures 6.14 to 6.16. These figures show that the normalized time weighted residual approach yields slightly less errors than the classical approach. For the particular nonlinear problem considered in this section, the two distinct approaches perform almost equally. However, it is to be noted that, for other types of nonlinear applications, the normalized time weighted residual approach outperforms the classical counterpart such as those observations seen in the context of structural dynamics problem [41–44] and the radiation heat transfer (Section 6.2). Therefore, in a general sense, the normalized time weighted residual approach can be seen to serve as a good approach to naturally extend the use of time integration methods, that were originally developed for linear systems, for use in nonlinear dynamic problems.

For completeness of the analysis, we explicitly demonstrate here the rate of convergence in time (i.e., the order of accuracy) of the GS4-1 framework employing the normalized time weighted residual approach for this particular problem. For

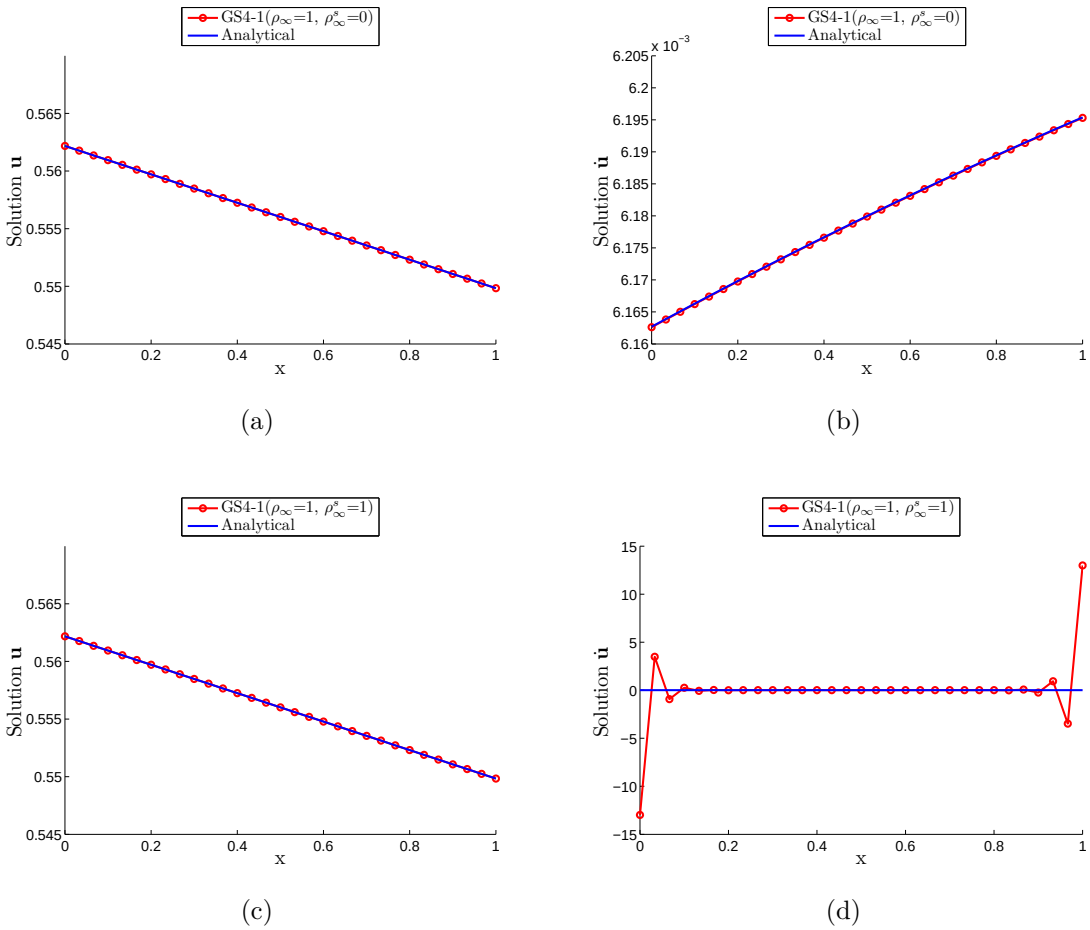
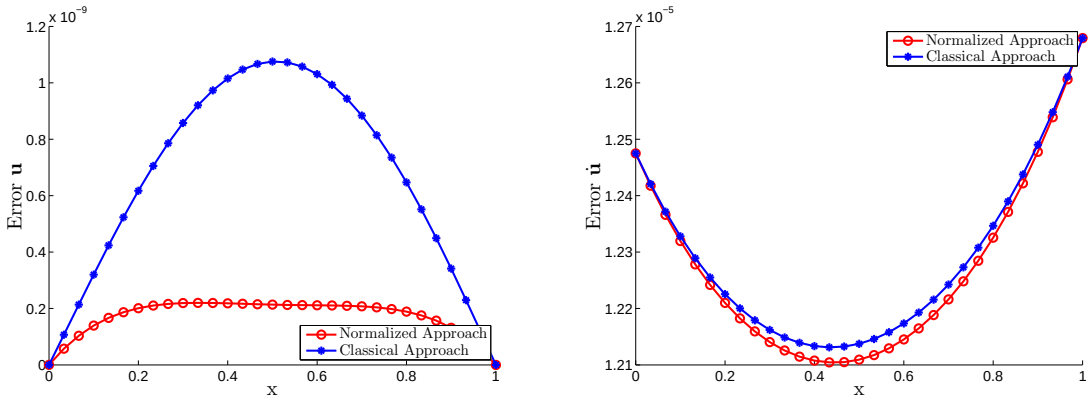
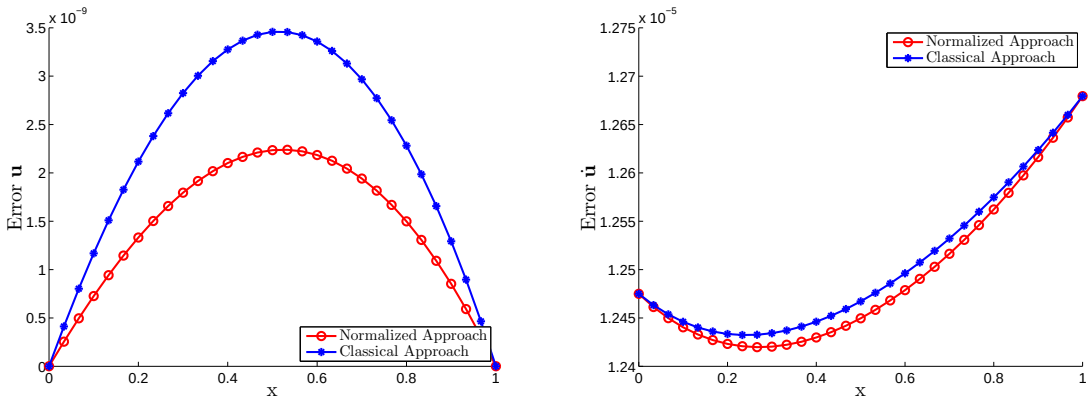


Figure 6.13: Plot of u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) as a function of x for a specific time of $t = 10$ with $\Delta t = 1$ and generated by: (i) GS4-1($\rho_\infty = 1, \rho^s_\infty = 0$) i.e., the case with the selective control feature, and (ii) GS4-1($\rho_\infty = 1, \rho^s_\infty = 1$) i.e., the case without the selective control feature, employing the normalized time weighted residual method (Section 6.3.4)

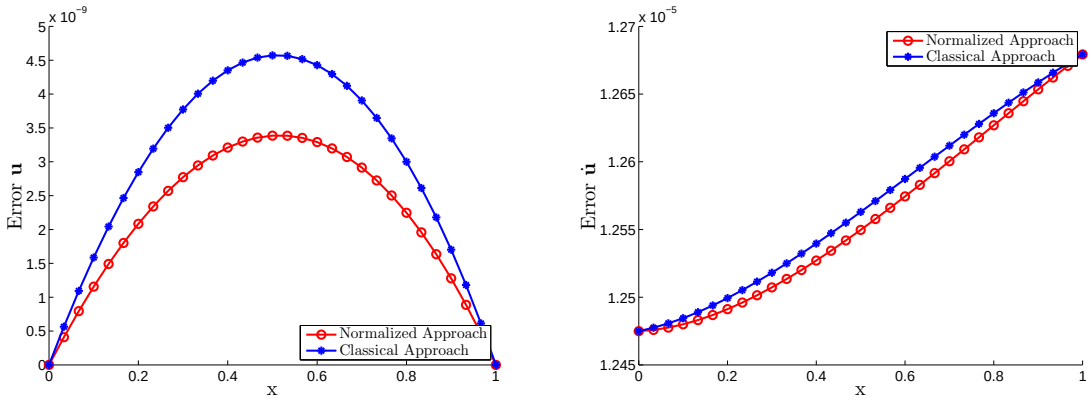


(a) $\rho_\infty = 1$ (b) $\rho_\infty = 1$

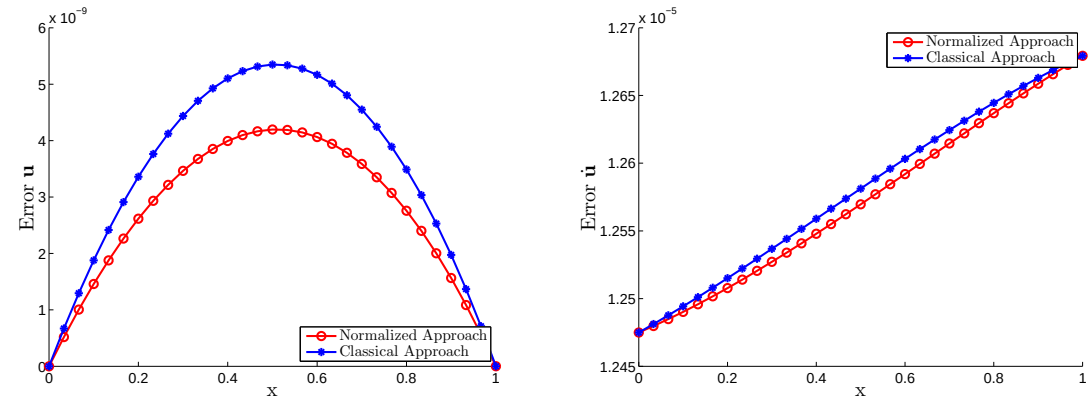


(c) $\rho_\infty = 0.9$ (d) $\rho_\infty = 0.9$

Figure 6.14: Comparison of errors in u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) using GS4-1 framework with the selective control feature (i.e., $\rho^s_\infty = 0$) between the different nonlinear treatments: (i) classical time weighted residual approach (Section 6.3.3), and (ii) normalized time weighted residual approach (Section 6.3.4), for ρ_∞ values of 0.9 and 1

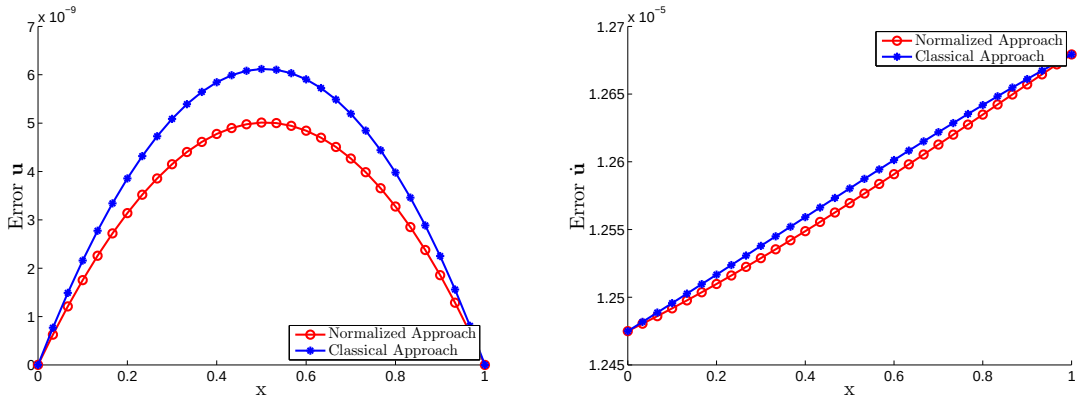


(a) $\rho_\infty = 0.8$ (b) $\rho_\infty = 0.8$

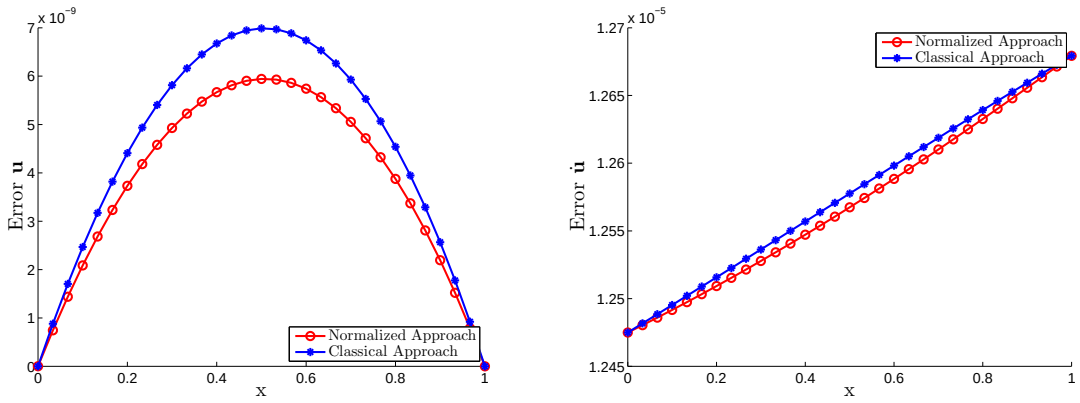


(c) $\rho_\infty = 0.7$ (d) $\rho_\infty = 0.7$

Figure 6.15: Comparison of errors in u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) using GS4-1 framework with the selective control feature (i.e., $\rho^s_\infty = 0$) between the different nonlinear treatments: (i) classical time weighted residual approach (Section 6.3.3), and (ii) normalized time weighted residual approach (Section 6.3.4), for ρ_∞ values of 0.7 and 0.8



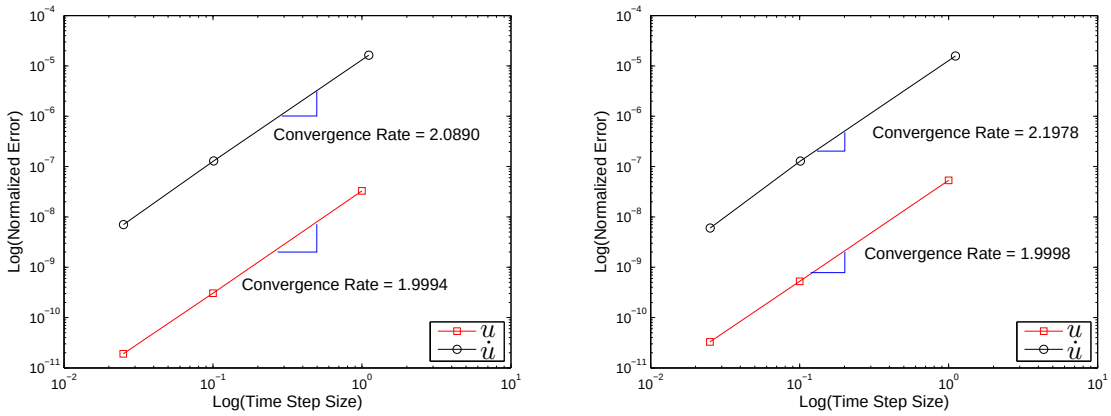
(a) $\rho_\infty = 0.6$ (b) $\rho_\infty = 0.6$



(c) $\rho_\infty = 0.5$ (d) $\rho_\infty = 0.5$

Figure 6.16: Comparison of errors in u and $\dot{u}$ (for the 1D Burgers equation, Section 6.3.6.1) using GS4-1 framework with the selective control feature (i.e., $\rho^s_\infty = 0$) between the different nonlinear treatments: (i) classical time weighted residual approach (Section 6.3.3), and (ii) normalized time weighted residual approach (Section 6.3.4), for ρ_∞ values of 0.5 and 0.6

this purpose, we choose the algorithms defined by: (a) $\rho_\infty = 1$, $\rho_\infty^s = 0$, and (b) $\rho_\infty = 0.6$, $\rho_\infty^s = 0$, and show the convergence plots in Figure 6.17. The total number of time steps (N_i, see equation (4.5.45)) used to construct the plots are chosen as 10000, 500, 200, and 100, where the solutions generated with $N_i = 10000$ are used as the reference to calculate the errors for the construction of the convergence plots. Meanwhile, the end time at which the solutions are evaluated is $t_{end} = 5$. In contructing these convergence plots, we use the standard convergence plot for u and the time level aligned convergence plot for $\dot{u}$, which is the proper construction of the convergence plot as discussed previously. Figure 6.17 demonstrates that these algorithms are second order accurate, hence the consistency.



(a) (b)

Figure 6.17: Convergence plot (for the 1D Burgers equation, Section 6.3.6.1) of the GS4-1 framework employing the normalized time weighted residual approach (Section 6.3.4) with: (a) $\rho_\infty = 1$, $\rho_\infty^s = 0$, and (b) $\rho_\infty = 0.6$, $\rho_\infty^s = 0$, both of which using the standard convergence plot for u and the time level aligned convergence plot for $\dot{u}$

6.3.6.2 2D Burgers Equation

Next, a two dimensional case is of interest to validate the above arguments and to provide consistent observations. The problem has the following Dirichlet type boundary and initial conditions (in dimensionless form)

$$
\begin{aligned}
u_x(0,y,t) &= \frac{3}{4} - \frac{1}{4[1+\exp((4y-t)Re/32)]} \\
u_x(x,0,t) &= \frac{3}{4} - \frac{1}{4[1+\exp((-4x-t)Re/32)]} \\
u_x(1,y,t) &= \frac{3}{4} - \frac{1}{4[1+\exp((-4+4y-t)Re/32)]} \\
u_x(x,1,t) &= \frac{3}{4} - \frac{1}{4[1+\exp((-4x+4-t)Re/32)]} \\
u_x(x,y,0) &= \frac{3}{4} - \frac{1}{4[1+\exp((-4x+4y)Re/32)]}
\end{aligned}
\tag{6.3.179}
$$

and

$$
\begin{aligned}
u_y(0,y,t) &= \frac{3}{4} + \frac{1}{4[1+\exp((4y-t)Re/32)]} \\
u_y(x,0,t) &= \frac{3}{4} + \frac{1}{4[1+\exp((-4x-t)Re/32)]} \\
u_y(1,y,t) &= \frac{3}{4} + \frac{1}{4[1+\exp((-4+4y-t)Re/32)]} \\
u_y(x,1,t) &= \frac{3}{4} + \frac{1}{4[1+\exp((-4x+4-t)Re/32)]} \\
u_y(x,y,0) &= \frac{3}{4} + \frac{1}{4[1+\exp((-4x+4y)Re/32)]}
\end{aligned}
\tag{6.3.180}
$$

where the exact solution is given by [79] (in dimensionless form)

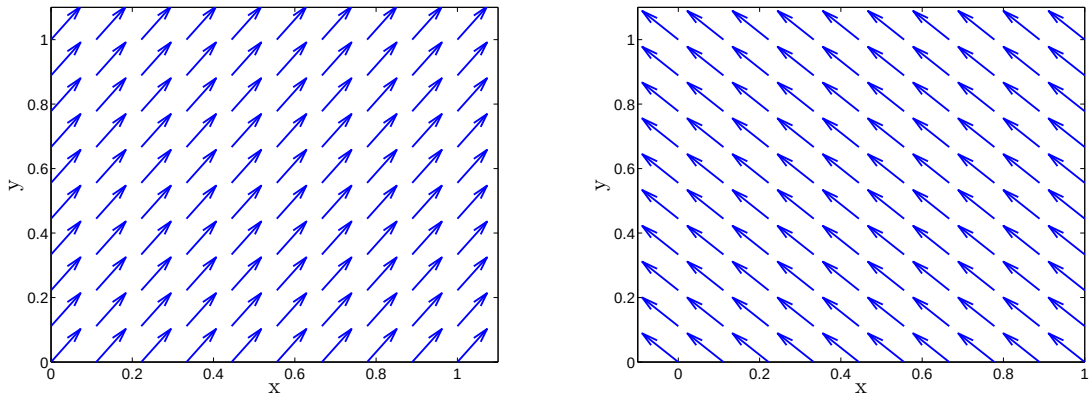
$$
\begin{aligned}
u_x(x,y,t) &= \frac{3}{4} - \frac{1}{4[1+\exp((-4x+4y-t)Re/32)]} \\
u_y(x,y,t) &= \frac{3}{4} + \frac{1}{4[1+\exp((-4x+4y-t)Re/32)]}
\end{aligned}
\tag{6.3.181}
$$

For this problem, we again choose $Re = 0.1$ (to eliminate any need for upwinding in the spatial discretization procedure). Since it is a 2D problem, we use 2D linear elements with typical element width $2b$ and height $2h$, whose element shape function is given by

$$
\begin{aligned}
\psi^{(e)} &= [\psi_1 \quad \psi_2 \quad \psi_3 \quad \psi_4] \\
\psi_1 &= \frac{(b-x)(h-y)}{4bh} \\
\psi_2 &= \frac{(b+x)(h-y)}{4bh} \\
\psi_3 &= \frac{(b+x)(h+y)}{4bh} \\
\psi_4 &= \frac{(b-x)(h+y)}{4bh}
\end{aligned}
\tag{6.3.182}
$$

We discretize the spatial domain using 10x10 elements and use the Galerkin FEM to discretize the spatial domain. For this problem, we again investigate whether the same algorithmic properties of the GS4-1 framework with the selective control feature observed in the previous numerical examples also hold in this case. For this purpose, we solve the problem using the two cases (i.e., the GS4-1 framework with and without the selective control feature) as the basic primitive algorithms to march the solutions in time, employing the normalized time weighted residual approach described in Section 6.3.4 with a time step size $\Delta t = 1$ and an end time $t = 10$. The analytical solutions at this end time, as given by equation (6.3.181), are shown in Figure 6.18. Meanwhile, the convergence criterion (tolerance) for the Newton-Raphson iteration is $tol = 1 \times 10^{-12}$.

To compare the performance of the two cases, we look at the solutions of the primary variable ($\mathbf{u}$) and its time derivative ($\dot{\mathbf{u}}$) as a function of time for a specific node at $x = 0.1$ and $y = 0.1$. For this purpose, we again choose $\rho_\infty = 1$ since the numerical solution of the primary variable for this particular problem is expected to be smooth even without imposing any numerical dissipation (due to the reason cited previously for the 1D example). Figure 6.19 shows the analytical solutions as given by equation (6.3.181). Meanwhile, Figure 6.20 shows the numerical solutions of $\mathbf{u} = (u_x, u_y)$ and $\dot{\mathbf{u}} = (\dot{u}_x, \dot{u}_y)$ as a function of time for this node generated by the two cases. From Figures 6.20(a) and 6.20(c), we can see that both cases yield good results for the two components of the velocity vector. However, the case without the selective control feature results in oscillations in the numerical solutions of both



(a) $\mathbf{u}$ (b) $\dot{\mathbf{u}}$

Figure 6.18: Vector plot of analytical solutions ($\mathbf{u}$ and $\dot{\mathbf{u}}$) (for the 2D Burgers equation, Section 6.3.6.2) at $t = 10$ as given by equation (6.3.181)

components of $\dot{\mathbf{u}}$, whereas the case with such a feature yields good agreement with the analytical solutions for the same ρ_∞ value as seen in Figures 6.20(b) and 6.20(d). These observations are consistent with those previously seen in all the previous numerical examples. Therefore, this 2D problem illustrates the consistency of the argument made previously, i.e., that the GS4-1 framework with the selective control feature has the advantage over the past and existing developments (in which the numerical dissipation is of limited control) through its ability to generate physically representative results for both the primary variable and its time derivative via such a new feature.

In addition, next, we determine the errors in $\mathbf{u}$ and $\dot{\mathbf{u}}$ (as defined in equation (6.3.178)) generated by the two cases for the whole spatial domain at a specific time of $t = 10$ and using a time step size of $\Delta t = 1$. For this purpose, we again employ the normalized time weighted residual approach (Section 6.3.4) while ρ_∞ is varied from 1 (nondissipative/zero damping) to 0 (maximal damping) in decrements of 0.1. The results are shown in Tables 6.3.1 and 6.3.2. As seen in Table 6.3.1, the two cases yield satisfactory results for the solution of $\mathbf{u}$ with only very small errors. Again, we can conclude that the two cases perform well in generating solutions of $\mathbf{u}$. Turning attention to the results of $\dot{\mathbf{u}}$, Table 6.3.2 shows that the case without the selective control feature yields very large errors

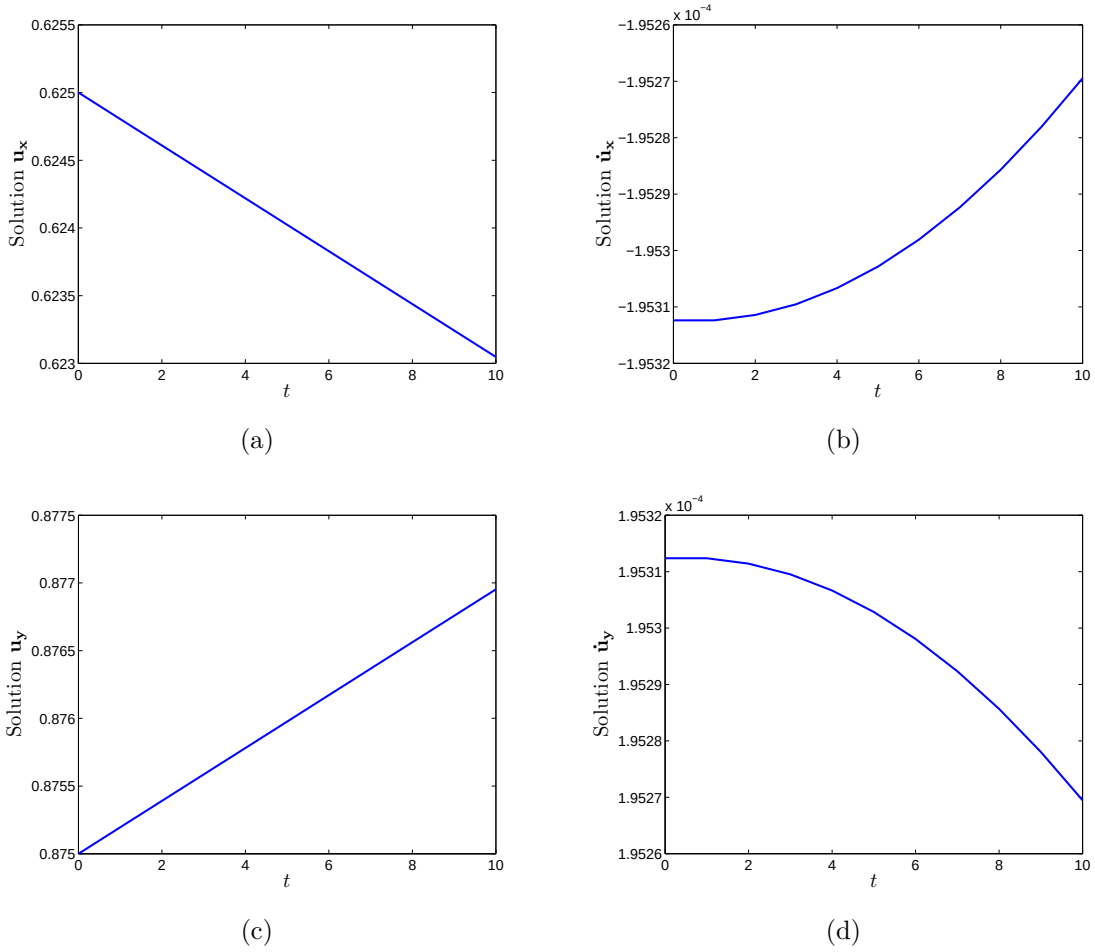


Figure 6.19: Plot of analytical solutions (u_x, u_y, $\dot{u}_x$ and $\dot{u}_y$) (for the 2D Burgers equation, Section 6.3.6.2) as a function of time for a specific node at $x = 0.1$ and $y = 0.1$ as given by equation (6.3.181)

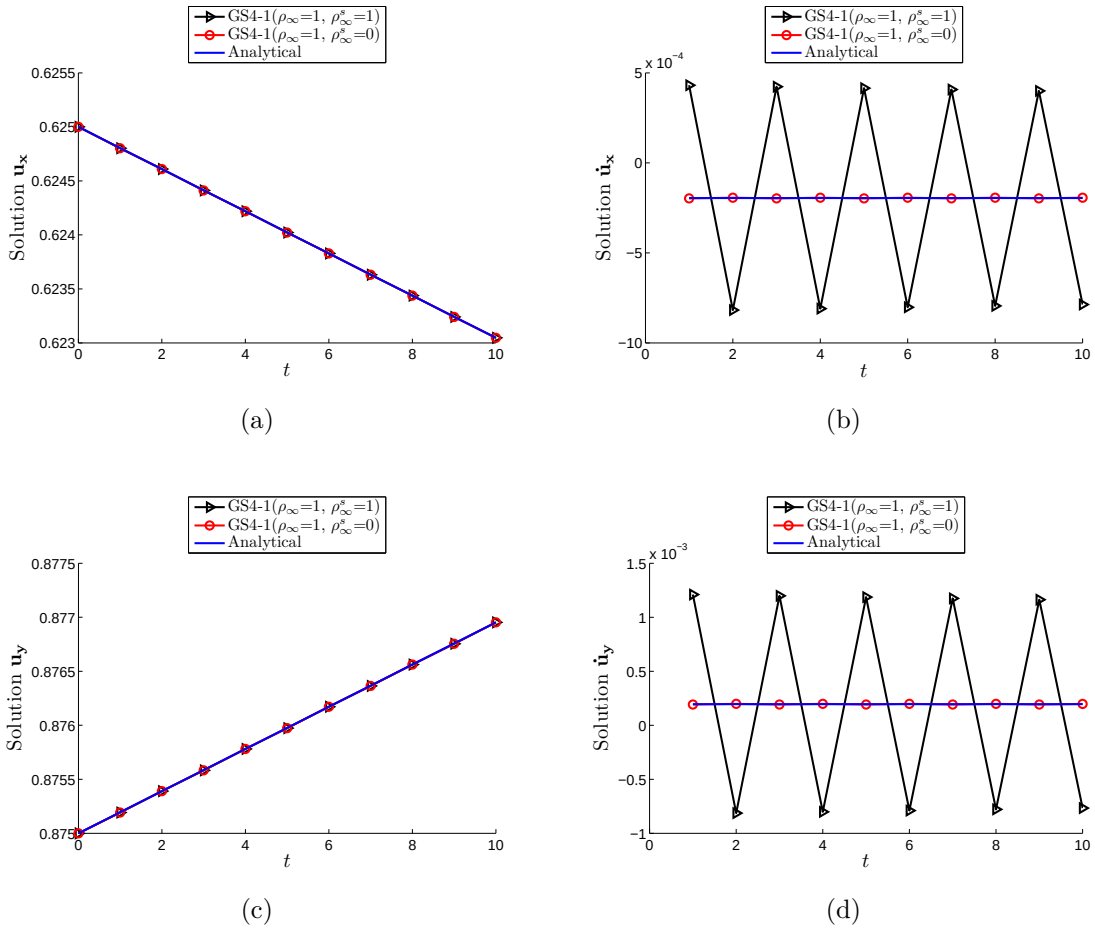


Figure 6.20: Plot of numerical solutions (u_x, u_y, $\dot{u}_x$ and $\dot{u}_y$) (for the 2D Burgers equation, Section 6.3.6.2) as a function of time for a specific node ($x = 0.1$, $y = 0.1$) generated by: (i) GS4-1($\rho_\infty = 1, \rho^s_\infty = 0$) i.e., with the selective control feature, and (ii) GS4-1($\rho_\infty = \rho^s_\infty = 1$), i.e., the case without the selective control feature, employing the normalized time weighted residual approach described in Section 6.3.4

especially at minimal numerical dissipation (ρ_∞ approaches 1). These errors are significantly reduced to an acceptable level by the GS4-1 framework when the selective control feature is enabled. This is again a repeated observation as seen in the previous numerical examples considered in this thesis, hence demonstrates the consistency of the argument. In this regard, we can state that the GS4-1 framework with the selective control feature is a definite improvement as compared to the existing and past developments to-date in which such a feature is not available.

Additionally, an investigation on the performance of the two nonlinear approaches, employing the GS4-1 framework with the selective control feature as the basic primitive algorithm to march the solutions in time, reveals that the same trends observed for the one-dimensional problem (Section 6.3.6.1) are repeated for this two-dimensional case, hence the consistency. Finally, it is worth mentioning that, although the effect of the selective control feature on the primary variable is not observed in the examples considered in this section, such effect has been demonstrated in Section 5.4, where the new GS4-1 framework is applied to a convective heat transfer problem with a thermal shock on the boundary, in which case the numerical solution of the primary variable is oscillating and therefore, the effect of the selective control feature on this variable is apparent (note that this will again be demonstrated in the numerical examples discussed in the next chapter).

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	$1.9651\text{x}10^{-6}$	$1.9651\text{x}10^{-6}$	$1.5362\text{x}10^{-4}$	$1.5362\text{x}10^{-4}$
0.9	$4.3395\text{x}10^{-6}$	$7.0316\text{x}10^{-6}$	$3.7284\text{x}10^{-4}$	$6.2423\text{x}10^{-4}$
0.8	$7.3149\text{x}10^{-6}$	$8.9924\text{x}10^{-6}$	$6.4887\text{x}10^{-4}$	$8.0663\text{x}10^{-4}$
0.7	$8.9981\text{x}10^{-6}$	$9.6679\text{x}10^{-6}$	$8.0603\text{x}10^{-4}$	$8.6960\text{x}10^{-4}$
0.6	$9.6758\text{x}10^{-6}$	$9.8683\text{x}10^{-6}$	$8.6983\text{x}10^{-4}$	$8.8835\text{x}10^{-4}$
0.5	$9.8786\text{x}10^{-6}$	$9.9171\text{x}10^{-6}$	$8.8915\text{x}10^{-4}$	$8.9293\text{x}10^{-4}$
0.4	$9.9214\text{x}10^{-6}$	$9.9260\text{x}10^{-6}$	$8.9330\text{x}10^{-4}$	$8.9378\text{x}10^{-4}$
0.3	$9.9269\text{x}10^{-6}$	$9.9271\text{x}10^{-6}$	$8.9385\text{x}10^{-4}$	$8.9388\text{x}10^{-4}$
0.2	$9.9272\text{x}10^{-6}$	$9.9272\text{x}10^{-6}$	$8.9389\text{x}10^{-4}$	$8.9389\text{x}10^{-4}$
0.1	$9.9272\text{x}10^{-6}$	$9.9272\text{x}10^{-6}$	$8.9389\text{x}10^{-4}$	$8.9389\text{x}10^{-4}$
0	$9.9272\text{x}10^{-6}$	$9.9272\text{x}10^{-6}$	$8.9389\text{x}10^{-4}$	$8.9389\text{x}10^{-4}$

Table 6.3.1: Comparison of error in $\mathbf{u}$ (for the 2D Burgers equation, Section 6.3.6.2) between: (i) the case without the selective control feature (SCF), i.e., with $\rho_\infty^s = \rho_\infty$, and (ii) GS4-1 framework with the selective control feature, i.e., with $\rho_\infty^s = 0$, for ρ_∞ values ranging from 1 (nondissipative/zero damping) to 0 (maximal damping) in decrements of 0.1, employing the normalized time weighted residual approach (Section 6.3.4)

ρ_∞	**Max Error**		**Total Error**	
	without SCF	with SCF	without SCF	with SCF
1	$2.7785\text{x}10^{3}$	0.0723	$1.4047\text{x}10^{5}$	5.7736
0.9	968.8014	0.0278	$4.8976\text{x}10^{4}$	2.2233
0.8	298.3381	0.0096	$1.5081\text{x}10^{4}$	0.7687
0.7	78.4853	0.0029	$3.9672\text{x}10^{3}$	0.2322
0.6	16.8004	$7.2077\text{x}10^{-4}$	849.1277	0.0593
0.5	2.7133	$1.4145\text{x}10^{-4}$	137.1317	0.0126
0.4	0.2913	$3.7219\text{x}10^{-5}$	14.7253	0.0028
0.3	0.0164	$2.5607\text{x}10^{-5}$	0.8295	0.0014
0.2	$2.8438\text{x}10^{-4}$	$2.4719\text{x}10^{-5}$	0.0150	0.0012
0.1	$2.4511\text{x}10^{-5}$	$2.4701\text{x}10^{-5}$	0.0012	0.0012
0	$2.4701\text{x}10^{-5}$	$2.4701\text{x}10^{-5}$	0.0012	0.0012

Table 6.3.2: Comparison of error in $\dot{\mathbf{u}}$ (for the 2D Burgers equation, Section 6.3.6.2) between: (i) the case without the selective control feature (SCF), i.e., with $\rho_\infty^s = \rho_\infty$, and (ii) GS4-1 framework with the selective control feature, i.e., with $\rho_\infty^s = 0$, for ρ_∞ values ranging from 1 (nondissipative/zero damping) to 0 (maximal damping) in decrements of 0.1, employing the normalized time weighted residual approach (Section 6.3.4)

Chapter 7

The Relations of the New GS4-1 and the Existing GS4-2 Time Integration Frameworks: Development of an Isochronous Integration Framework (iIntegrator) for First/Second Order Dynamic Systems

7.1 Introduction

For a complete development of the new GS4-1 time integration framework, the research presented in this thesis also sought a possible way for the new framework to be naturally integrated into the existing GS4-2 framework (second order system). A precise understanding of the respective frameworks, and how the two are related to each other, has led to the development of an isochronous integration framework [59], which enables the use of the same computational framework for solving both first and second order dynamic systems without having to resort to the individual

framework hence the practicality in computational and implementation aspects. Therefore, this completes the whole developments of the time integration framework, and can be viewed as another essential part of the research leading to a significant breakthrough. This is the focus in this chapter.

In this regard, of fundamental interest are: 1) first order transient systems such as those encountered in parabolic heat conduction, first order hyperbolic systems such as fluid flow, etc., and 2) second order dynamic systems such as those encountered in hyperbolic heat conduction, hyperbolic second order systems such as elastodynamics and wave propagation, etc. After spatial discretization using methods such as finite differences, finite volumes, finite elements, and the like, the subsequent proper integration of the time continuous ordinary differential equations is extremely important. In particular, the physical quantities of interest may need to be preserved and/or the equations should be optimally integrated so that there is only a minimal numerical dissipation and a minimal order of algorithm overshoot. Additionally, shocks (if any) should be captured without too much dissipation and stiff problems should be solved successfully.

From the overviews of the existing methods to-date described in Chapter 2, *Literature Review*, it is worth highlighting the following:

1. For linear second order dynamic systems, the existing time integration methods developed by various researchers may appear as totally separate families of algorithms derived from altogether totally different numerical approximation techniques; however, the GS4-2 framework was designed such that it recovers these existing algorithms regardless of the original approach of how the algorithms were developed (both numerically dissipative and nondissipative).

2. More importantly, the GS4-2 framework contains the family of optimal controllable numerically dissipative algorithms with new desirable features. These new features, which are zero-order displacement overshoot and zero-order velocity overshoot, are clearly improvement to the previous developments. Additionally, such optimal algorithms yield only minimal numerical dissipation and numeri-

cal dispersion for damped dynamic systems, and also have the option of energy conserving for undamped dynamic systems.

3. For nonlinear second order dynamic systems, the GS4-2 framework implemented via the normalized time weighted residual approach yields two types of computational framework with different conservation properties (i) a family of algorithms that can naturally achieve the conservation of the energy and momentum (both linear and angular) without having to enforce energy constraints (which is in contrast to all past works to-date), and/or alternatively (ii) a family of algorithms that are symplectic and momentum (both linear and angular) conserving for practical use in long-term simulation.

4. For first order transient systems, not only that the newly developed GS4-1 framework can recover the existing single step single solve algorithms, but it could also suppress most of the numerical oscillations associated with the primary variable and its time derivative via the new selective control feature to yield physically representative solutions that correctly capture the dynamics of the problem completely, which is in contrast to previous and existing algorithms without such a feature.

In a nutshell, both the GS4-1 and GS4-2 frameworks can be seen as the best state of the art time integration methods for first and second order dynamic systems, respectively, under the class of single step single solve methods. For a complete development of the time integration frameworks, we wish to further extend the success of the GS4-1 and GS4-2 frameworks by integrating/linking the two towards developing a unified time integration framework that can be used to solve both first and second order dynamic systems, hereby termed the *isochronous integration framework* (*i*Integrator).

Additionally, the following describes the motivation for the development of the *i*Integrator, which is the crux of the present chapter. From the overviews of the existing time integration methods to-date described in Chapter 2, it is worth noticing that these methods are developed for either first or second order dynamic systems, respectively. This means that an analyst has to resort to this individual framework when solving either of these systems. It would be very convenient for the analyst,

especially when the computational procedure and implementation has to be done by himself, if there exists a single framework that can be used to solve both systems, leaving out the need to implement the separate/individual frameworks when there is a need to switch from one system to another. In this regard, Zienkiewicz and co-workers [12] have in the past developed the so-called SSpj method, which is a general time integration algorithm with a single step for use in second or first order transient systems, presented in a general universal form for all orders of approximation. Although this method may seem to meet the need for a unified time integration framework suitable for use in both systems, it is worth to note that such a method is not up-to-date and lacks in terms of the optimal algorithms not yet available at the time of its development. Hence, the need for such an isochronous integration framework containing updated and current state-of-the-art optimal algorithms remains.

This has motivated us in this research to seek a possible way for the newly developed GS4-1 framework to be naturally integrated into the existing GS4-2 framework. If this could be accomplished, the outcome would therefore be of a novel breakthrough which completes the whole developments in the time integration field, and fills in the need for such an isochronous integration framework consisting of updated optimal algorithms that can be used to solve both first and second order systems without having to resort to the individual framework hence the practicality in computational and implementation aspects. This goal requires a precise understanding of the respective GS4-1 and GS4-2 frameworks, a thorough research of how the two are related to each other, and a key idea of how the relations can lead to the development of such an isochronous integration framework.

The isochronous integration framework presented here contains

1. For first order transient systems: the GS4-1 framework with its desirable features including second order accuracy, zero-order overshoot, unconditional stability, and controllable numerical dissipation with the new selective control feature.

2. For second order dynamic systems: the GS4-2 framework with its desirable

features including second order accuracy, unconditional stability and with the options of:

(a) conservation of properties (such as energy, symplectic, linear and angular momentum) for undamped system, and

(b) optimal controllable numerically dissipative algorithms with desirable features (zero-order displacement overshoot and zero-order velocity overshoot with only a minimal numerical dissipation and numerical dispersion) for damped or stiff systems.

Therefore, this is clearly a significant improvement to the state-of-the-art.

7.2 Reviews of GS4-2 and GS4-1 Frameworks

The key observations leading to the formalism of the *i*Integrator rise from a precise understanding of the respective frameworks, and how the two are related to each other, which further leads to the understanding of how the existing GS4-2 framework can be consistently adapted to yield the GS4-1 framework allowing for its use in both first and second order dynamic systems as an *i*Integrator. Therefore, it is important that we first review in this section the algorithmic structure of the individual GS4-2 and GS4-1 frameworks (note that although the algorithmic structure of the GS4-1 framework has been presented in Chapter 3, it is briefly repeated here to enhance the clarity of the descriptions and discussions regarding the relations between the two individual frameworks).

7.2.1 GS4-2 Framework for Time Integration of Second Order Dynamic System

Consider the semi-discretized linear second order dynamic system of the following governing equation and initial conditions

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{F}(t) \tag{7.2.1}$$

$$\mathbf{u}(0) = \mathbf{u}_0$$
$$\dot{\mathbf{u}}(0) = \dot{\mathbf{u}}_0$$

The above governing equation can be represented in the temporally discrete form by the GS4-2 framework as follows

$$\mathbf{M}\widetilde{\ddot{\mathbf{u}}} + \mathbf{C}\widetilde{\dot{\mathbf{u}}} + \mathbf{K}\widetilde{\mathbf{u}} = \widetilde{\mathbf{F}} \tag{7.2.2}$$

where

$$\widetilde{\ddot{\mathbf{u}}} = \ddot{\mathbf{u}}_n + \Lambda_6 W_1 \Delta\ddot{\mathbf{u}} \tag{7.2.3}$$
$$\widetilde{\dot{\mathbf{u}}} = \dot{\mathbf{u}}_n + \Lambda_4 W_1 \Delta t\ddot{\mathbf{u}}_n + \Lambda_5 W_2 \Delta t\Delta\ddot{\mathbf{u}} \tag{7.2.4}$$
$$\widetilde{\mathbf{u}} = \mathbf{u}_n + \Lambda_1 W_1 \Delta t\dot{\mathbf{u}}_n + \Lambda_2 W_2 \Delta t^2\ddot{\mathbf{u}}_n + \Lambda_3 W_3 \Delta t^2\Delta\ddot{\mathbf{u}} \tag{7.2.5}$$
$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{7.2.6}$$

Moreover, it is worth recalling that the GS4-2 framework can be divided into two sub-frameworks based on the overshoot behaviour: (1) the U0 family of algorithms, in which the displacement overshoot is zero-order (while the velocity overshoot is either zero or first order), and (2) the V0 family of algorithms, in which the velocity overshoot is zero-order (while the displacement overshoot is either zero or first order). For clarity, we provide the details of these two sub-frameworks separately as follows

Algorithm 7
The U0 Family of GS4-2 Algorithms for Second Order Dynamic System

Given $\mathbf{u}_n$, $\dot{\mathbf{u}}_n$, *and* $\ddot{\mathbf{u}}_n$, *one can find* $\mathbf{u}_{n+1}$, $\dot{\mathbf{u}}_{n+1}$, *and* $\ddot{\mathbf{u}}_{n+1}$ *by first solving for* $\Delta\ddot{\mathbf{u}}$ *from*

$$\begin{aligned}(\Lambda_6 W_1\mathbf{M} + \Lambda_5 W_2\Delta t\mathbf{C} + \Lambda_3 W_3\Delta t^2\mathbf{K})\Delta\ddot{\mathbf{u}} = &-\mathbf{M}\ddot{\mathbf{u}}_n - \mathbf{C}(\dot{\mathbf{u}}_n + \Lambda_4 W_1\Delta t\ddot{\mathbf{u}}_n)\\ &-\mathbf{K}(\mathbf{u}_n + \Lambda_1 W_1\Delta t\dot{\mathbf{u}}_n + \Lambda_2 W_2\Delta t^2\ddot{\mathbf{u}}_n)\\ &+\mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n)\end{aligned} \tag{7.2.7}$$

followed by updating the variables at the end of each time level as follows

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1 \dot{\mathbf{u}}_n \Delta t + \lambda_2 \ddot{\mathbf{u}}_n \Delta t^2 + \lambda_3 \Delta \ddot{\mathbf{u}} \Delta t^2 \tag{7.2.8}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4 \ddot{\mathbf{u}}_n \Delta t + \lambda_5 \Delta \ddot{\mathbf{u}} \Delta t^2 \tag{7.2.9}$$

$$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + \Delta \ddot{\mathbf{u}} \tag{7.2.10}$$

where

$$\begin{aligned}
\Lambda_1 W_1 &= \frac{1}{1+\rho_\infty^s}, \quad \lambda_1 = 1 \\
\Lambda_2 W_2 &= \frac{1}{2(1+\rho_\infty^s)}, \quad \lambda_2 = 1/2 \\
\Lambda_3 W_3 &= \frac{1}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})(1+\rho_\infty^s)}, \quad \lambda_3 = \frac{1}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})} \\
\Lambda_4 W_1 &= \frac{1}{1+\rho_\infty^s}, \quad \lambda_4 = 1 \\
\Lambda_5 W_2 &= \frac{3+\rho_\infty^{min}+\rho_\infty^{max}-\rho_\infty^{min}\rho_\infty^{max}}{2(1+\rho_\infty^{min})(1+\rho_\infty^{max})(1+\rho_\infty^s)}, \quad \lambda_5 = \frac{3+\rho_\infty^{min}+\rho_\infty^{max}-\rho_\infty^{min}\rho_\infty^{max}}{2(1+\rho_\infty^{min})(1+\rho_\infty^{max})} \\
\Lambda_6 W_1 &= \frac{2+\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^s-\rho_\infty^{min}\rho_\infty^{max}\rho_\infty^s}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})(1+\rho_\infty^s)} \\
W_1 &= \frac{1}{1+\rho_\infty^s}
\end{aligned} \tag{7.2.11}$$

The user control parameters (ρ_∞^{min}, ρ_∞^{max}, and ρ_∞^s) are the principal and spurious roots at the high frequency limit satisfying the following relation [27]

$$0 \leq \rho_\infty^s \leq \rho_\infty^{min} \leq \rho_\infty^{max} \leq 1 \tag{7.2.12}$$

Algorithm 8

The V0 Family of GS4-2 Algorithms for Second Order Dynamic System

Given $\mathbf{u}_n$, $\dot{\mathbf{u}}_n$, and $\ddot{\mathbf{u}}_n$, one can find $\mathbf{u}_{n+1}$, $\dot{\mathbf{u}}_{n+1}$, and $\ddot{\mathbf{u}}_{n+1}$ by first solving for $\Delta\ddot{\mathbf{u}}$ from equation (7.2.7), followed by updating the variables at the end of each time level using equations (7.2.8) to (7.2.10). Meanwhile, the parameters for this family are defined

in terms of the principal and spurious roots as follows

$$
\begin{aligned}
\Lambda_1 W_1 &= \frac{3+\rho_\infty^{min}+\rho_\infty^{max}-\rho_\infty^{min}\rho_\infty^{max}}{2(1+\rho_\infty^{min})(1+\rho_\infty^{max})}, \quad \lambda_1 = 1 \\
\Lambda_2 W_2 &= \frac{1}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})} \quad \lambda_2 = 1/2 \\
\Lambda_3 W_3 &= \frac{1}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})(1+\rho_\infty^{s})}, \quad \lambda_3 = \frac{1}{2(1+\rho_\infty^{s})} \\
\Lambda_4 W_1 &= \frac{3+\rho_\infty^{min}+\rho_\infty^{max}-\rho_\infty^{min}\rho_\infty^{max}}{2(1+\rho_\infty^{min})(1+\rho_\infty^{max})}, \quad \lambda_4 = 1 \\
\Lambda_5 W_2 &= \frac{2}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})(1+\rho_\infty^{s})}, \quad \lambda_5 = \frac{1}{1+\rho_\infty^{s}} \\
\Lambda_6 W_1 &= \frac{2+\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^{s}-\rho_\infty^{min}\rho_\infty^{max}\rho_\infty^{s}}{(1+\rho_\infty^{min})(1+\rho_\infty^{max})(1+\rho_\infty^{s})}, \\
W_1 &= \frac{3+\rho_\infty^{min}+\rho_\infty^{max}-\rho_\infty^{min}\rho_\infty^{max}}{2(1+\rho_\infty^{min})(1+\rho_\infty^{max})}
\end{aligned}
\tag{7.2.13}
$$

For the two families of the GS4-2 framework, namely the U0 and V0 families, with certain choices of the algorithmic parameters, the framework recovers existing methods as given in Tables 7.2.1 and 7.2.2. All algorithms belonging to the U0 and V0 families of the GS4-2 framework are second order accurate, unconditionally stable with minimal numerical dissipation and dispersion, and exhibit: (1) zero-order displacement overshoot and no more than first order velocity overshoot behaviours for the U0 family, and (2) no more than first order displacement overshoot and zero-order velocity overshoot behaviours for the V0 family. More importantly, when selecting $\rho_\infty^{max} = 1$, both the U0 and V0 families of the GS4-2 framework exhibit no overshoot behaviour in both displacement and velocity; these are termed as (U0-V0) family of algorithms, which is a clear improvement to the past/existing controllable numerically dissipative algorithms including the method described in [26].

Additionally, applying the GS4-2 framework as the basic primitive algorithm to march the solutions in time for nonlinear second order dynamic systems (structural dynamics with Saint Vernant-Kirchhoff material model), implemented via the normalized time weighted residual approach yields computational frameworks that: (i) can naturally

achieve the conservation of the energy and momentum (both linear and angular) without having to enforce energy constraints which is in contrast to all past works to-date [41], and/or alternatively (ii) can yield algorithms that are symplectic and momentum (both linear and angular) conserving for practical use in long-term simulation [43, 44].

7.2.2 GS4-1 Framework for Time Integration of First Order Transient System

Consider the semi-discretized linear first order transient system of the following governing equation and initial condition

$$\mathbf{M}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{F}(t) \tag{7.2.14}$$

$$\mathbf{u}(0) = \mathbf{u}_0 \tag{7.2.15}$$

The above governing equation can be represented in the temporally discrete form by the GS4-1 framework as follows

$$\mathbf{M}\widetilde{\dot{\mathbf{u}}} + \mathbf{K}\widetilde{\mathbf{u}} = \widetilde{\mathbf{F}} \tag{7.2.16}$$

where

$$\widetilde{\dot{\mathbf{u}}} = \dot{\mathbf{u}}_n + \Lambda_6 W_1 \Delta\dot{\mathbf{u}} \tag{7.2.17}$$

$$\widetilde{\mathbf{u}} = \mathbf{u}_n + \Lambda_4 W_1 \Delta t \dot{\mathbf{u}}_n + \Lambda_5 W_2 \Delta t \Delta\dot{\mathbf{u}} \tag{7.2.18}$$

$$\widetilde{\mathbf{F}} = \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \tag{7.2.19}$$

Therefore, one can solve such a problem using the GS4-1 framework as follows

Algorithm 9

The GS4-1 Framework for First Order Transient System

Given $\mathbf{u}_n$ *and* $\dot{\mathbf{u}}_n$*, one can find* $\mathbf{u}_{n+1}$ *and* $\dot{\mathbf{u}}_{n+1}$ *by first solving for* $\Delta\dot{\mathbf{u}}$ *from*

No.	Parameters	Algorithms
1	$\rho_\infty^{min} = \rho_\infty^{max} = \rho_\infty^s = 1$	Mid-point rule
2	$\rho_\infty^{min} = \rho_\infty^{max} = 1$, $\rho_\infty^s = 0$	Newmark method [10]
3	$\rho_\infty^{max} = 1$ and $\rho_\infty^{min} = \rho_\infty^s$	optimal algorithms with zero-order displacement and zero-order velocity overshoot behaviours and yield only minimal numerical dissipation and dispersion for damped dynamic systems (U0-V0$_{optimal}$) [16]
4	$\rho_\infty^{max} = 1$ and $\rho_\infty^s = 0$	discontinuous acceleration zero-order displacement and zero-order velocity overshoot behaviours (U0-V0$_{DA}$) family of algorithms [16].
5	$\rho_\infty^{max} = 1$ and $\rho_\infty^s = \frac{1-\rho_\infty^{min}\rho_\infty^{max}}{\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^{min}\rho_\infty^{max}}$	continuous acceleration zero-order displacement and zero-order velocity overshoot behaviours (U0-V0$_{CA}$) family of algorithms [16]
6	$\rho_\infty^{max} = 1$, $\rho_\infty^s \neq \rho_\infty^{min}$, $\rho_\infty^s \neq 0$ and $\rho_\infty^s \neq \frac{1-\rho_\infty^{min}\rho_\infty^{max}}{\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^{min}\rho_\infty^{max}}$	family of optimal algorithms with a zero-order displacement and zero-order velocity overshoot behaviours presented in [27]
7	$\rho_\infty^{min} = \rho_\infty^{max} = \rho_\infty^s$	optimal numerically dissipative family of algorithms with zero-order displacement and first-order velocity overshoot behaviours (U0-V1$_{optimal}$) [16], wherein the Generalized-α algorithm [26] is naturally recovered
8	$\rho_\infty^{min} = \rho_\infty^{max}$ and $\rho_\infty^s = 0$	discontinuous acceleration zero-order displacement and first-order velocity overshoot behaviours (U0-V1$_{DA}$) family of algorithms [16], wherein the WBZ algorithm [23] is naturally recovered
9	$\rho_\infty^{min} = \rho_\infty^{max}$ and $\rho_\infty^s = \frac{1-\rho_\infty^{min}\rho_\infty^{max}}{\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^{min}\rho_\infty^{max}}$	continuous acceleration zero-order displacement and first-order velocity overshoot behaviours (U0-V1$_{CA}$) family of algorithms [16], wherein the HHT algorithm [21, 80, 70] is naturally recovered
10	Otherwise	new family of algorithms with zero-order displacement and first-order velocity overshoot behaviours [27].

Table 7.2.1: The new and existing algorithms contained in the U0 family of the GS4-2 framework

No.	Parameters	Algorithms
1	$\rho_\infty^{min} = \rho_\infty^{max} = \rho_\infty^s = 1$	Mid-point rule
2	$\rho_\infty^{min} = \rho_\infty^{max} = 1$, and $\rho_\infty^s = 0$	Velocity Based Scheme [14, 53].
3	$\rho_\infty^{max} = 1$ and $\rho_\infty^{min} = \rho_\infty^s$	optimal algorithms with zero-order displacement and zero-order velocity overshoot behaviours and yield only minimal numerical dissipation and dispersion for damped dynamic systems (U0-V0$_{optimal}$) [16]
4	$\rho_\infty^{max} = 1$, and $\rho_\infty^s = 0$	discontinuous acceleration zero-order displacement and zero-order velocity overshoot behaviours (U0-V0$_{DA}$) family of algorithms [16].
5	$\rho_\infty^{max} = 1$, and $\rho_\infty^s = \frac{1-\rho_\infty^{min}\rho_\infty^{max}}{\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^{min}\rho_\infty^{max}}$	continuous acceleration zero-order displacement and zero-order velocity overshoot behaviours (U0-V0$_{CA}$) family of algorithms [16]
6	$\rho_\infty^{max} = 1$, $\rho_\infty^s \neq \rho_\infty^{min}$, $\rho_\infty^s \neq 0$, and $\rho_\infty^s \neq \frac{1-\rho_\infty^{min}\rho_\infty^{max}}{\rho_\infty^{min}+\rho_\infty^{max}+\rho_\infty^{min}\rho_\infty^{max}}$	family of optimal algorithms with a zero-order displacement and zero-order velocity overshoot behaviours presented in [27].
7	Otherwise	a new family of algorithms with first-order displacement and zero-order velocity overshoot behaviours [27].

Table 7.2.2: The new and existing algorithms contained in the V0 family of the GS4-2 framework

$$\begin{aligned}(\Lambda_6 W_1\mathbf{M} + \Lambda_5 W_2 \Delta t\mathbf{K})\Delta\dot{\mathbf{u}} = &-\mathbf{M}\dot{\mathbf{u}}_n - \mathbf{K}(\mathbf{u}_n + \Lambda_4 W_1 \Delta t\dot{\mathbf{u}}_n) \\ &+ \mathbf{F}_n + W_1(\mathbf{F}_{n+1} - \mathbf{F}_n) \end{aligned} \tag{7.2.20}$$

followed by updating the variables at the end of each time level as follows

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_4\dot{\mathbf{u}}_n\Delta t + \lambda_5\Delta\dot{\mathbf{u}}\Delta t \tag{7.2.21}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \Delta\dot{\mathbf{u}} \tag{7.2.22}$$

where

$$\begin{aligned}
\Lambda_6 W_1 &= \frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)(1+\rho_\infty^s)}, \quad W_1 = \frac{1}{1+\rho_\infty} \\
\Lambda_5 W_2 &= \frac{1}{(1+\rho_\infty)(1+\rho_\infty^s)}, \quad \lambda_5 = \frac{1}{1+\rho_\infty^s} \\
\Lambda_4 W_1 &= \frac{1}{1+\rho_\infty}, \quad \lambda_4 = 1
\end{aligned} \tag{7.2.23}$$

The user control parameters (ρ_∞ and ρ_∞^s) are the principal and spurious roots at the high frequency limit satisfying the following relation

$$0 \leq \rho_\infty^s \leq \rho_\infty \leq 1 \tag{7.2.24}$$

The corresponding GS4-1 framework in another form of representation is also available and presented in Chapter 3. All algorithms in the GS4-1 framework are second-order accurate, unconditional stable, zero-order overshoot, and controllable numerically dissipative, with the option of the selective control feature, which is a practically useful new feature to yield physically representative solutions of both variables, as described and demonstrated in Chapters 3 to 6. The existing and new algorithms contained in this framework is given in Table 7.2.3.

No.	Parameters	Algorithms
1	$\rho_\infty = \rho_\infty^s = 1$	Crank-Nicolson method [60]
2	$\rho_\infty = \rho_\infty^s = 0$	Gear's method [63]
3	$\rho_\infty = \rho_\infty^s$	existing algorithms without selective control feature [62]
4	$\rho_\infty \neq \rho_\infty^s$	new algorithms with selective control feature [66]

Table 7.2.3: The new and existing algorithms contained in the GS4-1 framework

In this chapter, our primary aim is to demonstrate that the GS4-2 framework can be consistently adapted to yield the GS4-1 framework, allowing for its use in both first and second order systems as an *i*Integrator. Before presenting how this can be achieved, it is important to highlight the relations between the two individual frameworks, which is the necessary ingredients for the formalism of the *i*Integrator.

7.3 The Relations between the GS4-2 and GS4-1 Frameworks

Close observations and comparisons of equations (7.2.3)-(7.2.6) and equations (7.2.17)-(7.2.19) indicate that the algorithmic variables of the GS4-2 and GS4-1 frameworks are related as follows

- $\widetilde{\ddot{\mathbf{u}}}$ in GS4-2 (equation (7.2.3)) takes similar form as $\widetilde{\dot{\mathbf{u}}}$ in GS4-1 (equation (7.2.17))
- $\widetilde{\dot{\mathbf{u}}}$ in GS4-2 (equation (7.2.4)) takes similar form as $\widetilde{\mathbf{u}}$ in GS4-1 (equation (7.2.18))
- $\widetilde{\mathbf{F}}$ in GS4-2 (equation (7.2.6)) takes similar form as $\widetilde{\mathbf{F}}$ in GS4-1 (equation (7.2.19))

Furthermore, if we let $\rho_\infty^{max} = 1$ and $\rho_\infty^{min} = \rho_\infty$ in the GS4-2 framework, it is easy to verify that the parameters $(\Lambda_i W_j, \lambda_k)$ appearing in the GS4-2 framework become those of the GS4-1 framework (however, note that for the U0 family of GS4-2 framework, it is further required that $\rho_\infty^{min} = \rho_\infty^s$). This means that by assigning $\rho_\infty^{max} = 1$ and $\rho_\infty^{min} = \rho_\infty$,

- $\widetilde{\ddot{\mathbf{u}}}$ in GS4-2 (equation (7.2.3)) is exactly $\widetilde{\dot{\mathbf{u}}}$ in GS4-1 (equation (7.2.17))
- $\widetilde{\dot{\mathbf{u}}}$ in GS4-2 (equation (7.2.4)) is exactly $\widetilde{\mathbf{u}}$ in GS4-1 (equation (7.2.18))
- $\widetilde{\mathbf{F}}$ in GS4-2 (equation (7.2.6)) is exactly $\widetilde{\mathbf{F}}$ in GS4-1 (equation (7.2.19))

This provides one important information: the GS4-2 framework that was originally developed for solving second order dynamic systems, can also be used to solve first order transient systems by properly assigning $\rho_\infty^{max} = 1$ and $\rho_\infty^{min} = \rho_\infty$ as discussed above. Furthermore, these assignments have to be complemented with the following additional assignments on the matrices so as to preserve the actual governing equation of the first order transient system

- Assign $\mathbf{M}$ in the GS4-2 framework as $\mathbf{M}$ in the first order transient system

- Assign **C** in the GS4-2 framework as **K** in the first order transient system
- Set **K** in the GS4-2 framework to equal **0** [1]
- Assign **F** in the GS4-2 framework as **F** in the first order transient system

The above can be viewed as the requirements in using the GS4-2 framework as the time integration solver for the first order transient systems. For convenience, such requirements will be summarized in the following section, and hereby termed as the '*Adapting Formula*', through which the GS4-2 framework can be properly adapted to consistently yield the GS4-1 framework, allowing for the formalism of the isochronous integration framework that can be used to solve both first and second order dynamic systems. The Adapting Formula, which relates the two individual frameworks consistently, explains how one can appropriately switch from one framework to the other. This important formula is discussed next.

7.4 The Formalism of *i*Integrator via the Adapting Formula

Figure 7.1 illustrates the concept of the *i*Integrator and, in particular, how the GS4-2 framework can be properly adapted to consistently yield the GS4-1 framework via the Adapting Formula. Using this formula, we have essentially interpreted the $(n+1)^{st}$ order derivative and its corresponding coefficient matrix as the n^{th} order derivative and its corresponding coefficient matrix. Performing the set of assignments described by the Adapting Formula yields the GS4-1 family of algorithms exactly as presented in Section 7.2.2 (also described in detailed in Chapter 3). That is, this formula is carefully formulated using the knowledge of how the two frameworks are related to each other, such that the GS4-2 framework can be correctly adapted to yield the corresponding GS4-1 framework while preserving the physical interpretation and criteria of the algorithmic parameters in both frameworks. It correctly explains the relations or links that exist between the two frameworks, which subsequently allows for the formalism of the isochronous integration framework (*i*Integrator). It

[1] Note that because of this assignment, the variable **u** in the GS4-2 framework no longer participates in the solution procedure. Hence, this variable becomes a 'dummy variable' whose presence in the solution procedure is to be neglected. This way, the actual governing equation of the first order system is preserved and the GS4-2 framework consistently yields the GS4-1 framework

is therefore another significant contribution of the work presented in this thesis.

The Adapting Formula is given in Table 7.4.4. It is to be referred to as one wishes to use the GS4-2 framework for solving a first order transient system. Meanwhile, for use in second order dynamic system, the original GS4-2 framework can be used without any needs for adjustments. In this regard, we can view the GS4-2 framework as being the *i*Integrator, containing both the original GS4-2 framework and the newly developed GS4-1 framework. Although this may highlight the significant contribution of the GS4-2 framework alone, it is important to recall that the discovery of it being an isochronous integration framework had stemmed from the precise understanding of both the individual GS4-1 and GS4-2 frameworks and, in particular, of how the two frameworks are related to each other. This requires, in the first place, the successful designs and developments of the two individual frameworks. Therefore, the significance of the development of the GS4-1 framework, as presented in this thesis, remains.

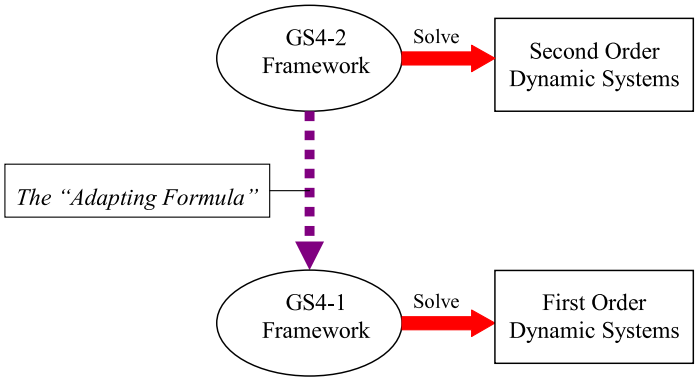


Figure 7.1: Illustration of the isochronous integration framework (*i*Integrator) concept

The physical interpretations of assigning the parameters ($\rho_\infty^{min}, \rho_\infty^{max}, \rho_\infty^{s}$) according to the Adapting Formula (Table 7.4.4) are as follows: The principal roots (ρ_∞^{min} and ρ_∞^{max}) in the GS4-2 framework are related to the principal root in the GS4-1 framework (ρ_∞). These principal roots control the numerical dissipation associated with the primary variables in both frameworks, respectively. Since the GS4-1 framework has only one principal root, this root is represented by the parameter ρ_∞^{min} in the GS4-2 framework, while it is required that $\rho_\infty^{max} = 1$ so as to satisfy

For matrices	Assign $\mathbf{M}$ in the GS4-2 framework as $\mathbf{M}$ in the first order system
	Assign $\mathbf{C}$ in the GS4-2 framework as $\mathbf{K}$ in the first order system
	Set $\mathbf{K}$ in the GS4-2 framework to equal $\mathbf{0}$
	Assign $\mathbf{F}$ in the GS4-2 framework as $\mathbf{F}$ in the first order system
For variables	Treat $\ddot{\mathbf{u}}$ in the GS4-2 framework as $\dot{\mathbf{u}}$ in the first order system
	Treat $\dot{\mathbf{u}}$ in the GS4-2 framework as $\mathbf{u}$ in the first order system
	Neglect $\mathbf{u}$ in the GS4-2 framework (i.e., dummy variable)
For parameters	Set ρ_∞^s in the GS4-2 framework as ρ_∞^s in the GS4-1 framework
	Set ρ_∞^{max} in the GS4-2 framework to equal 1
	Set ρ_∞^{min} in the GS4-2 framework as ρ_∞ in the GS4-1 framework
For U0 Family	Additionally require $\rho_\infty^s = \rho_\infty^{min} \in [0,1]$ and yield the GS4-1 framework without the selective control feature
For V0 Family	Choose $\rho_\infty^s \leq \rho_\infty^{min} \in [0,1]$ and yield the GS4-1 framework with/without the selective control feature

Table 7.4.4: The '*Adapting Formula*' for properly using the GS4-2 framework as the GS4-1 framework (see also Figure 7.1)

the required condition in the GS4-2 framework that $0 \leq \rho_\infty^s \leq \rho_\infty^{min} \leq \rho_\infty^{max} \leq 1$. Meanwhile, the spurious roots ρ_∞^s in both frameworks are consistently related to each other, and they control the numerical dissipation associated with the highest order time derivative variables in both framework, respectively.

Remark 7.4.1

1. *The requirement that $\rho_\infty^{max} = 1$ in the Adapting Formula is crucial for the recovery of the GS4-1 framework from the GS4-2 framework while preserving the physical meaning and criteria of the parameters in both frameworks as described above.*

2. *It is worthy to note that due to this requirement ($\rho_\infty^{max} = 1$), the Generalized-α method (which is contained in the U0 family of the GS4-2 framework) cannot readily be adapted to yield the corresponding algorithms for the first order transient systems since in this method, $\rho_\infty^{max} = \rho_\infty^{min} = \rho_\infty^s \neq 1$ (see Table 7.2.1). This highlights the disadvantage of the Generalized-α method, i.e., that it cannot be used as an isochronous integration framework in comparison to the GS4-2 framework. The same holds for the HHT-α and the WBZ methods.*

3. *Additionally, the Newmark method, which is also contained in the U0 family*

of the GS4-2 framework, also cannot be adapted to yield the corresponding algorithm for first order transient systems since in this method $\rho_\infty^s \neq \rho_\infty^{min}$ (for this method, although the requirement that $\rho_\infty^{max} = 1$ is satisfied, recall from Table 7.4.4 that for the U0 family of GS4-2 framework, the Adapting Formula requires that $\rho_\infty^s = \rho_\infty^{min}$).

4. *On the other hand, the Velocity Based Scheme, the U0-V0$_{optimal}$ algorithm, and the variations of the U0-V0 algorithms contained in the V0 family of the GS4-2 framework can readily be adapted for use in first order transient system since in this methods $\rho_\infty^{max} = 1$ (see Table 7.2.2 and also recall from Table 7.4.4 that for the V0 family of GS4-2 framework, the Adapting Formula does not require that $\rho_\infty^s = \rho_\infty^{min}$). More importantly, these methods recover algorithms in the GS4-1 framework with the selective control feature, which is a new feature not available in existing methods to-date for obtaining physically accurate and representative solutions of all variables.*

7.5 Numerical Illustrations: Classical Thermoelasticity Problems

We present in this section the application of the developed *i*Integrator to solve classical thermoelasticity problem which involves both the first and second order dynamic systems. The problem is described next.

7.5.1 Governing Equations

Interdisciplinary problems involving thermal sciences and structural dynamics, especially in hostile thermal environments, has long received considerable attention in widespread research activities [81]. The fusion of both fields of heat conduction in solids and continuum mechanics results in the field of dynamic thermoelasticity. The interdisciplinary nature and the complexity of the problem require robust, reliable, efficient and practical computational methods for interfacing the individual disciplines into a unified analysis. This is where the present *i*Integrator [59] fits in.

We shall first present the general theory of classical linear thermoelasticity problems. For homogeneous and isotropic continuum, the governing equations for the temperature and displacement fields are the following coupled differential equations [82]

$$\frac{E}{2(1+\nu)}\left(u_{i,\,kk}+\frac{1}{1-2\nu}u_{k,\,ki}\right)+\rho f_i-\rho\ddot{u}_i-\frac{E\alpha}{1-2\nu}T_{,i}=0, \tag{7.5.25}$$

$$\lambda T_{,kk}+r-\rho c_v\dot{T}-\frac{E\alpha T_0}{1-2\nu}\dot{u}_{k,\,k}=0, \tag{7.5.26}$$

where E is the Young's modulus; ν is Poisson's ratio; u_i and f_i are the Cartesian components of displacement and body force vector, respectively; ρ is the mass density; α is the coefficient of linear thermal expansion; T is the absolute temperature; λ is the thermal conductivity; r is the heat source; c_v is the specific heat at constant strain and T_0 is the reference temperature of the natural, stress-free state. Meanwhile, superposed dots $\dot{(\,)}$ and commas $(\,)_{,i}$ denote time differentiation and partial differential with respect to Cartesian co-ordinates $x_i (i = 1, 2, 3)$, respectively.

The equation of motion for the displacement field (equation (7.5.25)) is a hyperbolic, second order (in time) system. Meanwhile, the governing equation for the temperature field (equation (7.5.26)) is parabolic, first order (in time) system. Since the developed *i*Integrator presented here is meant to be used to solve the first and second order dynamic systems, such classical thermoelasticity problems governed by the above equations are well suited for consideration in this chapter to truly illustrate the method's ability to solve both systems. Although non-classical thermoelastic models frequently appear in the technical literature [81], the governing equations for the displacement and temperature fields in this non-classical category are both second order in time, hence are not appropriate for consideration here. The discussion from here onwards will therefore be focused on the classical thermoelasticity problems involving first and second order dynamic systems.

In the theoretical studies as well as engineering practice of thermoelasticity problems, simplifications to the above fully coupled equations are usually made, from which adequate results can be obtained relatively more easily. These simplifications involve neglecting the inertia term in the equation of motion (equation (7.5.25)) to arrive at a quasi-static model, and/or eliminating the coupling term from the heat conduction equation (equation (7.5.26)) to arrive at an uncoupled model. Only the latter

is of interest here due to the reason cited above. Furthermore, analytical solutions to the uncoupled classical thermoelastic model are available in the literatures, enabling the validation of the developed numerical method. For the uncoupled classical thermoelastic model, the heat conduction equation (equation (7.5.26)) becomes

$$\lambda T_{,kk} + r - \rho c_v \dot{T} = 0, \tag{7.5.27}$$

In particular, we consider here a thermal shock problem of a half-space, which is frequently used as a test example for thermoelastic problems due to its relative simplicity. Analytical solutions of such a problem have been reported since 1950, with the pioneer work due to Danilovskaya[2] as well as Sternberg and Chakravorty[3] [85], who were the first to give closed-form, uncoupled, dynamic solutions for the temperature and stress fields. The former, which involves a sudden, step-jump heating on the boundary, was later named after her as '*Danilovskaya's First Problem*', whereas the latter involves a ramp-type surface heating and is often referred to as the '*Sternberg and Chakravorty's Problem*'. In this work, these problems were considered and the statement of the problems is given next.

7.5.2 Statement of Problems

Take an (x, y, z) Cartesian coordinate system and consider a homogeneous, isotropic, thermoelastic solid occupying the half-space $x \geq 0$. Supposed that the solid is initially at rest, in stress-free state, at a uniform temperature of $T = T_0$. At time $t = 0^+$, however, the temperature at the boundary of the solid ($x = 0$) is changed from $T = T_0$ to $T = T_1$ by either a sudden, step-jump heating (for the Danilovskaya's First Problem) or a ramp-type surface heating (Sternberg and Chakravorty's Problem), and is maintained steadily at this value (i.e., T_1 is a stationary value). The boundary is free to move without any restrictions (i.e., is maintained stress-free). The temperature, displacement, and stress fields after the surface heating can be expressed as follows

[2]Note that the original work by Danilovskaya is described in [83]

[3]Meanwhile, the original work by Sternberg and Chakravorty is described in [84]

$$u_x = u_x(x,t), \quad u_y = 0, \quad u_z = 0, \quad T = T(x,t), \tag{7.5.28}$$

$$\sigma_x = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\frac{\partial u_x}{\partial x} - \frac{E\alpha}{1-2\nu}(T - T_0), \tag{7.5.29}$$

$$\sigma_y = \sigma_z = \frac{\nu}{1-\nu}\sigma_x - \frac{E\alpha}{1-\nu}(T - T_0) \tag{7.5.30}$$

An observation of equations (7.5.28) to (7.5.30) indicate that the determination of the temperature (T), x-direction displacement (u_x) and normal stress (σ_x) are of interest for a complete analysis. From the value of the temperature and normal stress, the stresses σ_y and σ_z can be determined from equation (7.5.30). For this case, and in the absence of body forces (f_i) and heat source (r), equations (7.5.25) and (7.5.27) can be expressed as

$$\frac{\partial^2 u_x}{\partial x^2} - \frac{(1+\nu)(1-2\nu)\rho}{(1-\nu)E}\frac{\partial^2 u_x}{\partial t^2} - \frac{(1+\nu)\alpha}{1-\nu}\frac{\partial T}{\partial x} = 0, \tag{7.5.31}$$

$$\frac{\partial^2 T}{\partial x^2} - \frac{\rho c_v}{\lambda}\frac{\partial T}{\partial t} = 0 \tag{7.5.32}$$

To complete the description of the problem, the associated initial and boundary conditions can be expressed as follows

$$\begin{aligned} &u_x(x,0) = 0, \quad \frac{\partial}{\partial t}u_x(x,0) = 0, \quad T(x,0) = 0, \\ &u_x(x \to \infty, t) \to 0, \quad \frac{\partial}{\partial x}u_x(x \to \infty, t) \to 0, \quad T(x \to \infty, t) \to T_0, \\ &\sigma_x(0,t) = 0, \quad T(0,t) = f(t) \end{aligned} \tag{7.5.33}$$

where, for the Danilovskaya First Problem (sudden heating case), $f(t)$ in equation (7.5.33) is given by

$$f(t) = T_0 + (T_1 - T_0)H(t) \tag{7.5.34}$$

while for the Sternberg-Chakravorty Problem (ramp-type surface heating), it is given by

$$f(t) = T_0 + \frac{(T_1 - T_0)t}{t_0}[H(t) - H(t - t_0)] + (T_1 - T_0)H(t - t_0) \tag{7.5.35}$$

where $H(t)$ is the Heaviside unit step-function and t_0 represents the time-lag for the surface temperature to reach the stationary value T_1. Now, we define the following parameters

$$\kappa = \frac{\lambda}{\rho c_v}, \quad c = \left[\frac{(1-\nu)E}{(1+\nu)(1-2\nu)\rho}\right]^{1/2} \tag{7.5.36}$$

$$\xi = \frac{cx}{\kappa}, \quad \tau = \frac{c^2 t}{\kappa} \tag{7.5.37}$$

where κ is the thermal diffusivity and c is the velocity of the isothermal elastic wave (first sound), while ξ and τ are the dimensionless space and time coordinates. Using these parameters, it is expedient to define dimensionless temperature, displacement and normal stress as follows

$$\theta = \frac{T - T_0}{T_0}, \quad u = \frac{(1-\nu)c}{\kappa(1+\nu)\alpha T_0}u_x, \quad \sigma = \frac{(1-2\nu)}{E\alpha T_0}\sigma_x \tag{7.5.38}$$

By the use of these dimensionless variables, the governing equations of the displacement and temperature fields can be represented in dimensionless form, together with the normal stress, as follows

$$\frac{\partial^2 u}{\partial \xi^2} - \frac{\partial^2 u}{\partial \tau^2} - \frac{\partial \theta}{\partial \xi} = 0 \tag{7.5.39}$$

$$\frac{\partial^2 \theta}{\partial \xi^2} - \frac{\partial \theta}{\partial \tau} = 0 \tag{7.5.40}$$

$$\sigma = \frac{\partial u}{\partial \xi} - \theta \tag{7.5.41}$$

or, in a more concise form

$$u'' - \ddot{u} - \theta' = 0 \tag{7.5.42}$$

$$\theta'' - \dot{\theta} = 0 \tag{7.5.43}$$

$$\sigma = u' - \theta \tag{7.5.44}$$

the primes $()'$ and superposed dots $(\dot{})$ denote partial differentiations with respect to the dimensionless variables ξ and τ, respectively. Likewise, the intial and boundary conditions can be expressed in dimensionless forms as follows

$$u(\xi, 0) = 0, \quad \dot{u}(\xi, 0) = 0, \quad \theta(\xi, 0) = 0, \tag{7.5.45}$$

$$u(\xi \to \infty, \tau) \to 0, \quad u'(\xi \to \infty, \tau) \to 0, \quad \theta(\xi \to \infty, \tau) \to 0 \tag{7.5.46}$$

$$\sigma(0, \tau) = 0, \quad \theta(0, \tau) = \varphi(\tau) \tag{7.5.47}$$

For simplification purpose, we suppose that $T_1 = 2T_0$, in which case $\varphi(\tau)$ can be expressed for the Danilovskaya's First Problem as

$$\varphi(\tau) = H(\tau) \tag{7.5.48}$$

while for the Sternberg-Chakravorty boundary condition as

$$\varphi(\tau) = \frac{\tau}{\tau_0}[H(\tau) - H(\tau - \tau_0)] + H(\tau - \tau_0) \tag{7.5.49}$$

Equations (7.5.42) and (7.5.43) indicate that the equation of motion describing the displacement field is a second order dynamic system, while the heat conduction equation describing the temperature field is a first order transient system. Solving such

problem would require appropriate time integration solvers for both the second and first order transient systems, respectively. Therefore, this illustrative example serves well to demonstrate the applicability of the present isochronous integration framework. The descriptions of how this example (or any problems involving first and second order systems) can be solved effectively and practically using the present *i*Integrator will be presented in Section 7.5.4 where we discuss the time discretization of such problem. In the next section, we first present the spatial discretization procedures for the problem, employing the Finite Element Method.

7.5.3 Spatial Discretization by the Finite Element Method

Observation of the governing equations, equations (7.5.42) and (7.5.43), indicates that the equation of motion describing the displacement field (equation (7.5.42)) is dependent on the temperature field that is to be solved from the heat conduction equation (equation (7.5.43)). Therefore, in the following computational procedures, we first discuss the heat conduction model, followed by the discussion on the equation of motion describing the displacement field.

7.5.3.1 Heat Conduction Equation for the Temperature Field

To discretize the heat conduction equation using the Galerkin Finite Element Method, we apply the method of weighted residual to equation (7.5.43) for an arbitrary element (e) with N nodes bounded by $\Omega^{(e)}$ domain. The temperature field of the element is approximated by a linear combination of time dependent nodal temperatures $\theta_n^{(e)}(n = 1, ..., N)$ and time independent element shape functions $\psi_n^{(e)}(n = 1, ..., N)$

$$\theta^{(e)} = \sum_{n=1}^{N} \theta_n^{(e)} \psi_n^{(e)} \tag{7.5.50}$$

The shape function $\psi_n^{(e)}$ is used as the weighting function in the weighted residual equation. After use of equation (7.5.50), we obtain the following system of linear ordinary differential equations that is first order in time

$$\mathbf{M}^{(e)} \dot{\boldsymbol{\theta}}^{(e)} + \mathbf{K}^{(e)} \boldsymbol{\theta}^{(e)} = \mathbf{F}^{(e)} \tag{7.5.51}$$

where $\boldsymbol{\theta}^{(e)}$ is a vector of length N containing the nodal temperatures $\theta_n^{(e)}(n = 1, ...N)$, $\dot{\boldsymbol{\theta}}^{(e)}$ is the time derivative of $\boldsymbol{\theta}^{(e)}$, while $\mathbf{M}^{(e)}$ is the mass matrix of size N x N and is defined by ($p = 1, ..., N$ and $q = 1, ..., N$)

$$M^{(e)}(p, q) = \int_{\Omega^{(e)}} \psi_p^{(e)} \psi_q^{(e)} d\Omega \tag{7.5.52}$$

Furthermore, $\mathbf{K}^{(e)}$ is the stiffness matrix of size N x N and is defined by ($p = 1, ..., N$ and $q = 1, ..., N$)

$$K^{(e)}(p, q) = \int_{\Omega^{(e)}} \frac{d\psi_p^{(e)}}{d\xi} \frac{d\psi_q^{(e)}}{d\xi} d\Omega \tag{7.5.53}$$

Meanwhile, $\mathbf{F}^{(e)}$ is zero vector of length N. The system of ordinary differential equations (7.5.51) is consistently assembled for all elements in the spatial domain to yield a global system of linear ordinary differential equations that is first order in time. For this problem, we discretize the spatial domain using 2000 1D elements with a total length of 40 in the ξ direction to represent the half-space (i.e., $\xi \to \infty$).

7.5.3.2 Equation of Motion for the Displacement Field

Employing the Finite Element Method to equation (7.5.42) for an arbitrary element (e) with N nodes bounded by $\Omega^{(e)}$ domain, following the same procedure as done for the heat conduction equation (Section 7.5.3.1), we obtain for the displacement equation of motion the following linear ordinary differential equation that is second order in time

$$\mathbf{M}^{(e)} \ddot{\mathbf{u}}^{(e)} + \mathbf{K}^{(e)} \mathbf{u}^{(e)} = \mathbf{F}^{(e)} \tag{7.5.54}$$

where $\mathbf{u}^{(e)}$ is a vector of length N containing the nodal displacements $u_n^{(e)}(n = 1, ...N)$, $\ddot{\mathbf{u}}^{(e)}$ is the time derivative of $\mathbf{u}^{(e)}$, while $\mathbf{M}^{(e)}$ and $\mathbf{K}^{(e)}$ are the mass and stiffness matrices of size N x N defined by equations (7.5.52) and (7.5.53), respectively.

Meanwhile, for this model, $\mathbf{F}^{(e)}$ is the force vector of length N defined as ($p = 1, ..., N$ and $q = 1, ..., N$)

$$F^{(e)}(p) = \left(\int_{\Omega^{(e)}} \sum_{q=1}^{N} \frac{d\psi_p^{(e)}}{d\xi} \psi_q^{(e)} d\Omega \right) \boldsymbol{\theta}_p^{(e)} \tag{7.5.55}$$

where $\boldsymbol{\theta}_p^{(e)}$ is the nodal temperature value at node p ($p = 1, ..., N$) of this element to be obtained from solving equation (7.5.51) using the present *i*Integrator. The system of ordinary differential equations (7.5.54) is consistently assembled for all elements in the spatial domain to yield a system of linear ordinary differential equations that is second order in time, to be solved using the specially tailored *i*Integrator to effectively capture the problem physics. This is described next.

7.5.4 Time Discretization by the *i*Integrator

In this section, we demonstrate how the present *i*Integrator can be used to solve the considered thermoelasticity problems.

7.5.4.1 Danilovskaya's First Problem

The heat conduction model results in a system of ordinary differential equations that is first order in time (see equation (7.5.51)). For such a system, we recall from Section 7.4 that one needs to impose the Adapting Formula described in Table 7.4.4 to correctly adapt the GS4-2 framework to yield the corresponding GS4-1 framework suitable for solving such a first order transient system. For the particular heat conduction model considered in this section, the Adapting Formula can be imposed as follows.

From equation (7.5.51), we have the following global system (upon the assembly of all elements)

$$\mathbf{M}\dot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta} = \mathbf{F} \tag{7.5.56}$$

By referring to the Adapting Formula (Table 7.4.4), and in accordance with to equation (7.5.56), one performs the following assigments in the computational procedure

(input part) of the GS4-2 framework:

1. Assign $\mathbf{M}$ in the GS4-2 framework as $\mathbf{M}$ in equation (7.5.56)
2. Assign $\mathbf{C}$ in the GS4-2 framework as $\mathbf{K}$ in equation (7.5.56)
3. Set $\mathbf{K}$ in the GS4-2 framework to equal $\mathbf{0}$
4. Assign $\mathbf{F}$ in the GS4-2 framework as $\mathbf{F}$ in equation (7.5.56)

Additionally, we treat the variables in the GS4-2 framework as follows

1. Treat $\ddot{\mathbf{u}}$ in the GS4-2 framework as $\dot{\boldsymbol{\theta}}$ in equation (7.5.56)
2. Treat $\dot{\mathbf{u}}$ in the GS4-2 framework as $\boldsymbol{\theta}$ in equation (7.5.56)
3. Neglect $\mathbf{u}$ in the GS4-2 framework (i.e., dummy variable)

Finally, we set the algorithmic parameters in the GS4-2 framework according to the Adapting Formula as follows

1. Set ρ_∞^s in the GS4-2 framework as ρ_∞^s in the GS4-1 framework (that controls the numerical dissipation of $\dot{\boldsymbol{\theta}}$ in equation (7.5.56))
2. Set $\rho_\infty^{max} = 1$ in the GS4-2 framework
3. Set ρ_∞^{min} in the GS4-2 framework as ρ_∞ in the GS4-1 framework (that controls numerical dissipation of $\boldsymbol{\theta}$ in equation (7.5.56))

This transforms the GS4-2 framework exactly to the GS4-1 framework suitable for solving equation (7.5.56) that governs the heat transfer model. This way, the original GS4-1 framework needs not be programmed directly. In other words, the computational implementation of the GS4-1 framework can make use of the same computational code as for the GS4-2 framework by performing the assignments

outlined above (i.e., as described by the Adapting Formula).[4] This demonstrates the practicality of the *i*Integrator, that enables the use of the same computational code to solve both first and second order systems.

We employ the *i*Integrator for solving the heat transfer model as described above (in which case the framework recovers the original GS4-1 framework) using a dimensionless time step size ($\Delta\tau$) of 0.1 and a dimensionless end time of $\tau = 2$ (see equation (7.5.37) for definition of the dimensionless time, τ). Additionally, we demonstrate the ability of the selective control feature inherent in the framework. As previously described, such a feature can be enabled by choosing $\rho_\infty^{min} \neq \rho_\infty^s$ (in the GS4-2/*i*Integrator), in which case the *i*Integrator recovers the GS4-1 framework with the selective control feature (i.e., with $\rho_\infty \neq \rho_\infty^s$). To achieve this, we choose the V0 family of the GS4-2 framework since the U0 family does not allow $\rho_\infty^{min} \neq \rho_\infty^s$ (see Table 7.4.4). On the other hand, when $\rho_\infty^{min} = \rho_\infty^s$ the framework recovers the GS4-1 framework without such a feature as in the existing/past methods. Figures 7.2 to 7.5 show the plots of θ and $\dot{\theta}$ as a function of time for a specific node at $\xi = 0.02$ employing the two cases, i.e., the GS4-1 framework with and without the selective control feature (see *Remark* 4.3.1) for ρ_∞ values of 1, 0.9, 0.8, and 0.7. It can be seen from these figures that the case without the selective control feature results in large numerical oscillations in the solutions of $\dot{\theta}$ for all the ρ_∞ values considered. Meanwhile, these oscillations can be easily reduced when the selective control feature is enabled (i.e., by choosing $\rho_\infty^s = 0$ via the selective control feature). In terms of the numerical solution of θ, these figures show that for a particular value of ρ_∞, both cases yield almost similar results (there is however a slight difference). However, the selective control feature does play role in improving the numerical solution of this variable. To see this clearly, additionally, we plot the error in θ generated by the two cases for the different ρ_∞ values considered. The error is defined as

$$Error = \left| \frac{Numerical - Exact}{Exact} \right| \times 100\% \tag{7.5.57}$$

[4]However, as previously highlighted, the development of the original GS4-1 framework as presented in this thesis is crucial since the development of the *i*Integrator requires the precise understanding of both the individual GS4-1 and GS4-2 frameworks and, in particular, of how the two are related to each other.

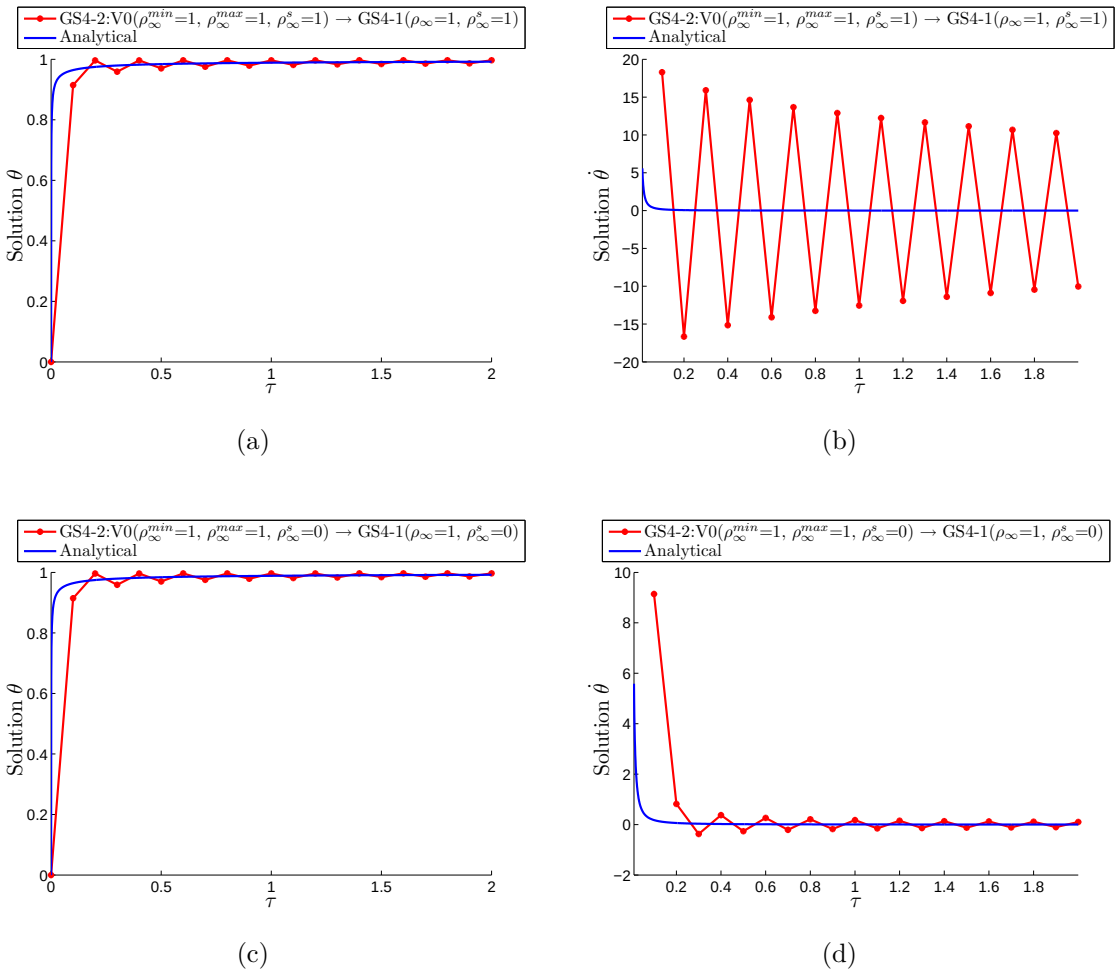


Figure 7.2: Plot of θ and $\dot{\theta}$ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of time for a specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 1$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 1, \rho_\infty^s = 0$) i.e., the case with selective control feature

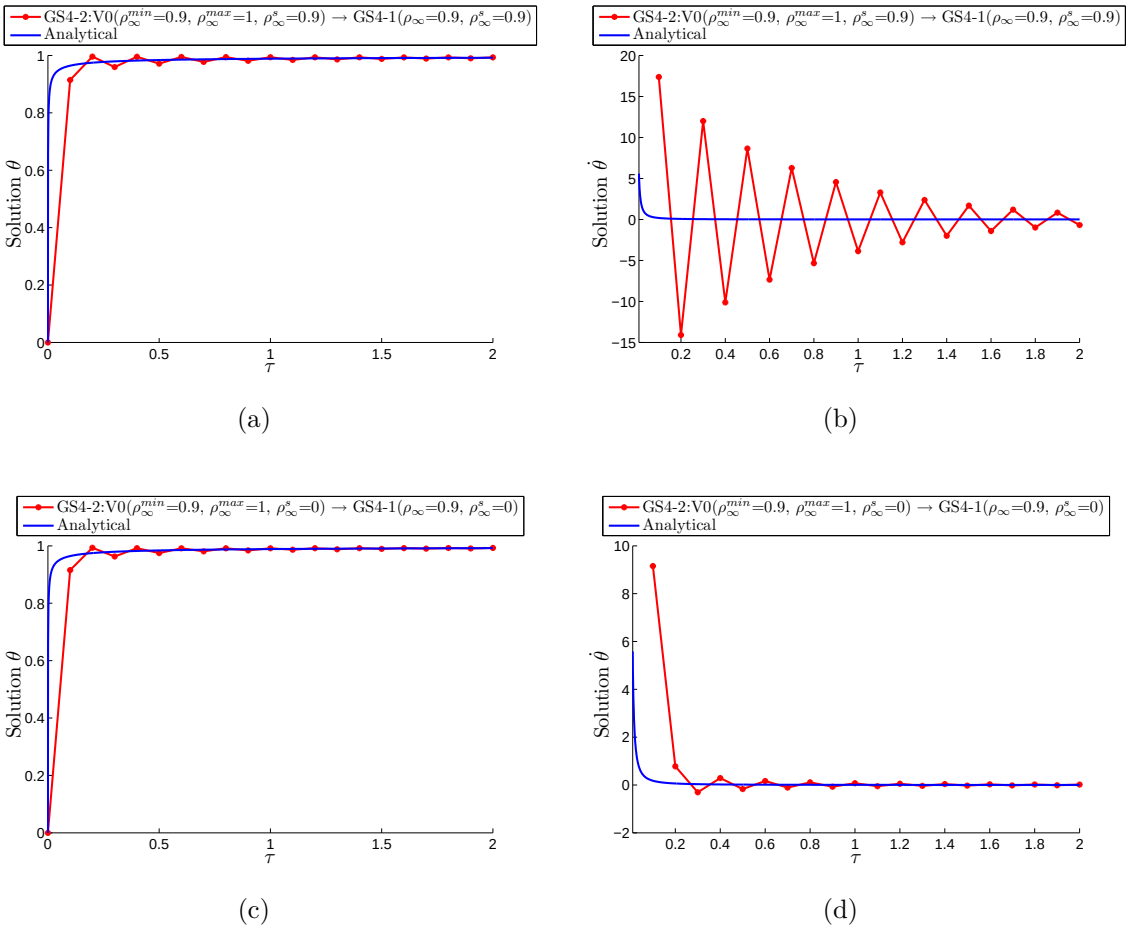


Figure 7.3: Plot of θ and $\dot{\theta}$ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of time for a specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 0.9$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 0.9, \rho_\infty^s = 0$) i.e., the case with selective control feature

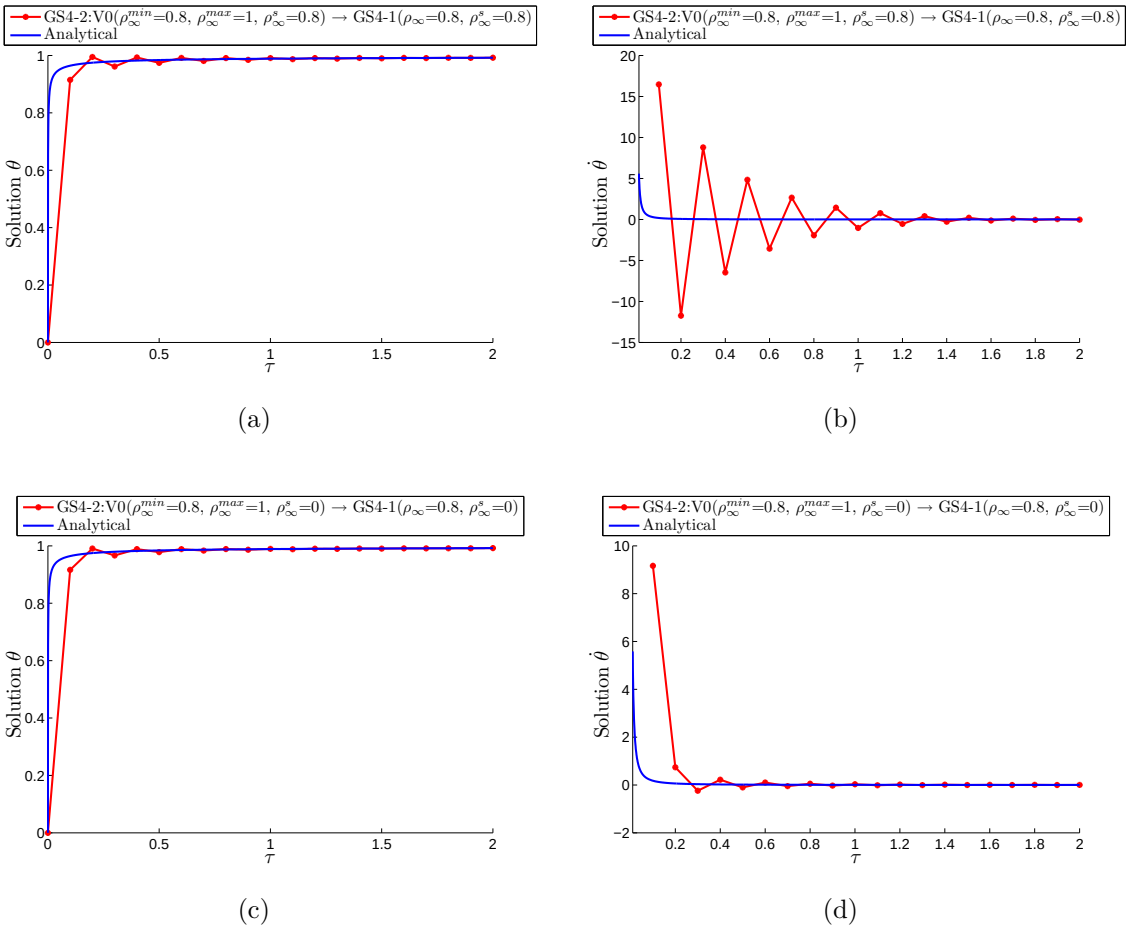


Figure 7.4: Plot of θ and $\dot{\theta}$ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of time for a specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 0.8$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 0.8, \rho_\infty^s = 0$) i.e., the case with selective control feature

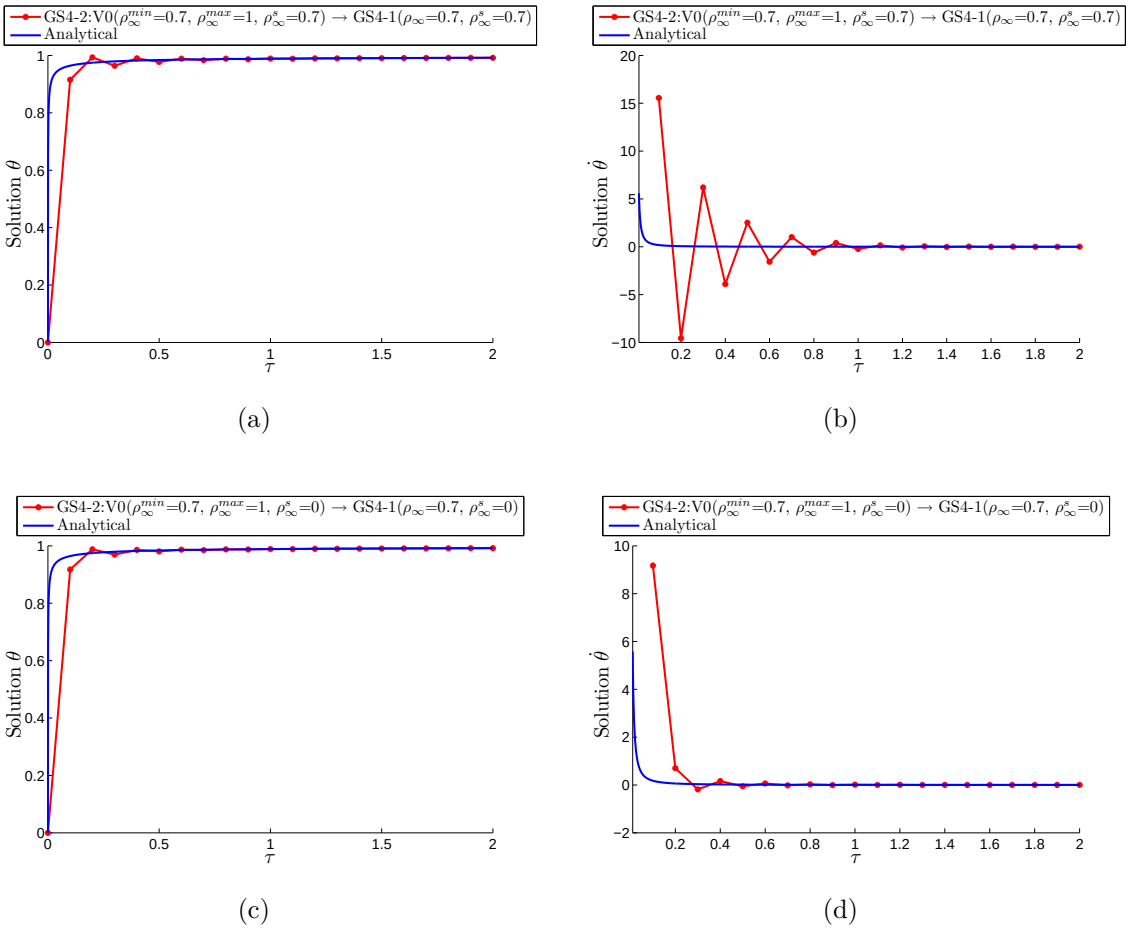


Figure 7.5: Plot of θ and $\dot{\theta}$ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of time for a specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 0.7$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 0.7, \rho_\infty^s = 0$) i.e., the case with selective control feature

Figure 7.6 shows the error in θ as a function of time for a specific node at $\xi = 0.02$ employing the two cases for ρ_∞ values of 1, 0.9, 0.8, and 0.7. Figure 7.6(a) indicates that, for $\rho_\infty = 1$, the error in θ generated by the two cases are the same. This is to be expected; as discussed before, the selection of ρ_∞=1 and any $\rho_\infty^s \in [0, 1]$ result in the same numerical solution of the primary variable (i.e., θ for this problem) (see Section 4.1.3). However, observation of Figures 7.6(b) to 7.6(d) reveals that the case with the selective control feature yields slightly less error in θ as compared to the case without such a feature. This shows that the selective control feature also plays role towards improving the accuracy of the numerical solution of the primary variable, in addition to yielding physically accurate solutions of the time derivative variable.

We next solve the system of ordinary differential equations describing the displacement field as given by equation (7.5.54), i.e., (upon the assembly of all elements)

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F} \tag{7.5.58}$$

In this case, the system is second order in time, and therefore we employ the *i*Integrator in its natural form without having to apply the Adapting Formula, in which case the framework recovers the original GS4-2 framework. Particularly, in accordance with equation (7.5.58), we do the following

1. Assign $\mathbf{M}$ in the GS4-2 framework as $\mathbf{M}$ in equation (7.5.58)
2. Set $\mathbf{C}$ in the GS4-2 framework to equal $\mathbf{0}$ [5]
3. Assign $\mathbf{K}$ in the GS4-2 framework as $\mathbf{K}$ in equation (7.5.58)
4. Assign $\mathbf{F}$ in the GS4-2 framework as $\mathbf{F}$ in equation (7.5.58)

Finally, we choose the values of the parameters in the GS4-2 framework as follows

1. Choose ρ_∞^s value in the GS4-2 framework that controls the numerical dissipation of the acceleration ($\ddot{\mathbf{u}}$)

[5] since equation (7.5.58) does not have the $\mathbf{C}$ component

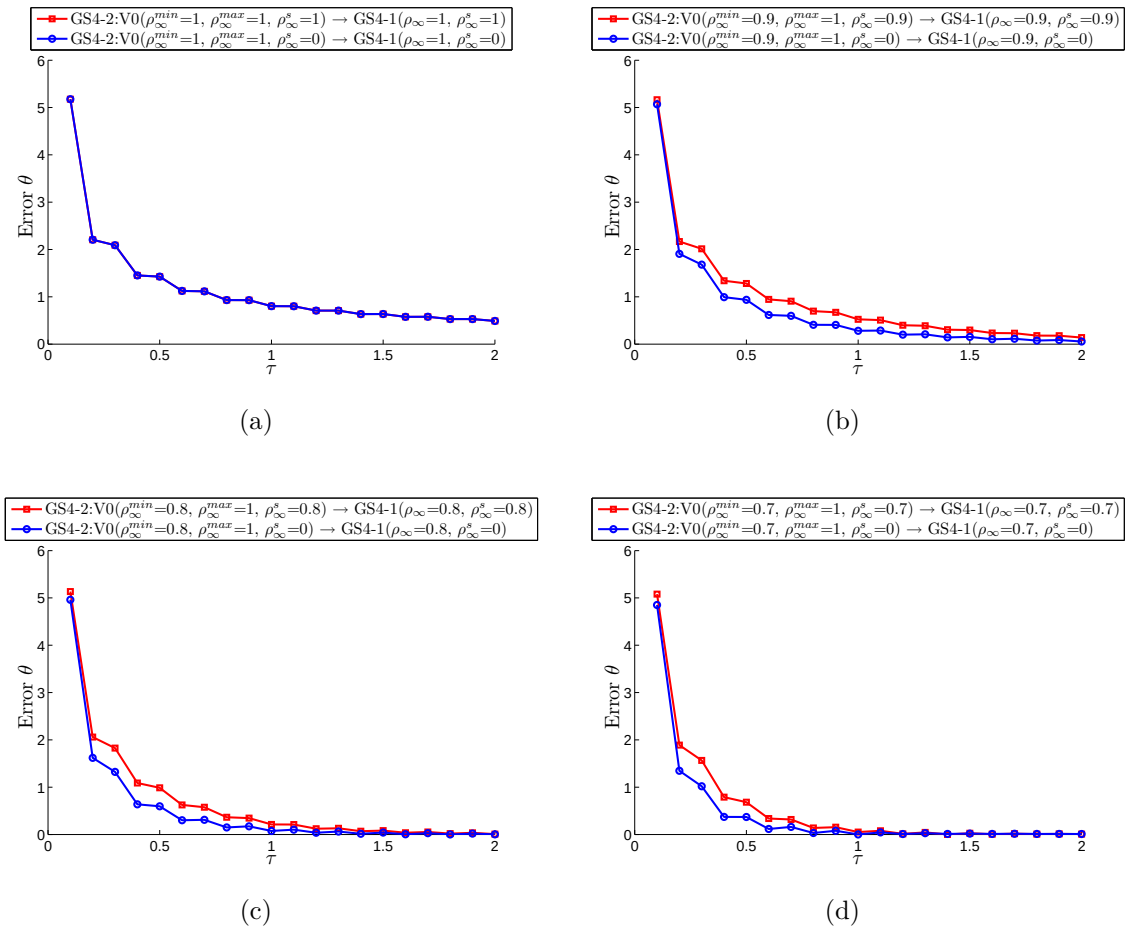


Figure 7.6: Plot of error in θ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of time for a specific node at $\xi = 0.02$ employing: (i) GS4-1 with $\rho_\infty^s = \rho_\infty$ i.e., the case without selective control feature, and (ii) GS4-1 with $\rho_\infty^s = 0$ i.e., the case with selective control feature, for $\rho_\infty = 1$, 0.9, 0.8, and 0.7

2. Choose ρ_∞^{max} value in the GS4-2 framework that controls the numerical dissipation of the velocity ($\dot{\mathbf{u}}$)

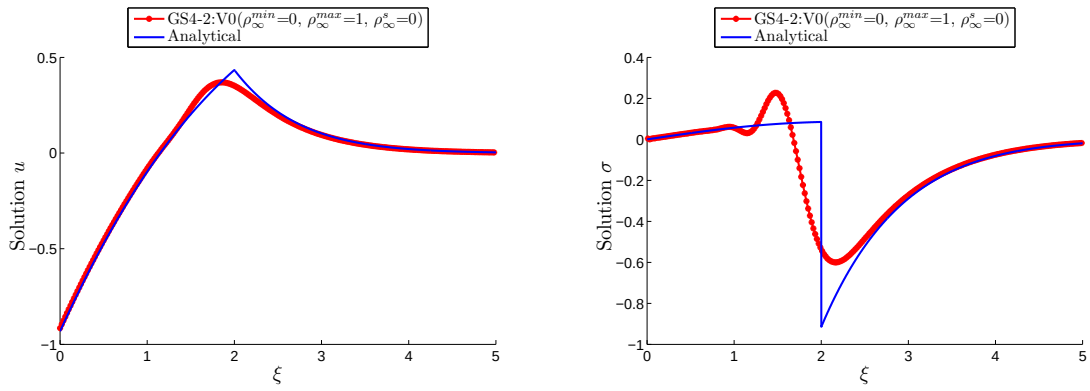
3. Choose ρ_∞^{min} value in the GS4-2 framework that controls the numerical dissipation of the displacement ($\mathbf{u}$)

At this point, it is worthy to recall that the primary objective in this chapter is to describe the *i*Integrator that can be used to solve both first and second order dynamic systems, and to demonstrate how to switch from one system to the other via the Adapting Formula. Consequently, we intentionally put aside the investigation and discussion on applying various algorithms in the GS4-2 framework for the second order dynamic system as this has been demonstrated in our previous works [16, 27, 41, 43, 44]. For this reason, we only employ here the U0-V0$_{optimal}$ defined by GS4-2:V0($\rho_\infty^{min} = 0$, $\rho_\infty^{max} = 1$, $\rho_\infty^s = 0$). Since the computational procedure to solve this system requires the solution of the heat conduction model (the first order transient system), we use the solution of θ generated by the GS4-1 framework with the selective control feature defined by $\rho_\infty = 0.7$ and $\rho_\infty^s = 0$ as this algorithm provides the least errors in both θ and $\dot{\theta}$ as shown above. The numerical results on the dimensionless displacement (u) and stress (σ) as a function of the spatial coordinate ξ at $\tau = 2$ are shown in Figures 7.7 and 7.8 for $\Delta\tau$ values of 0.1 and 0.01, respectively.[6] From these figures, it can be seen that the numerical solutions of these variables employing the U0-V0$_{optimal}$ are satisfactory.

7.5.4.2 Sternberg-Chakravorty's Problem

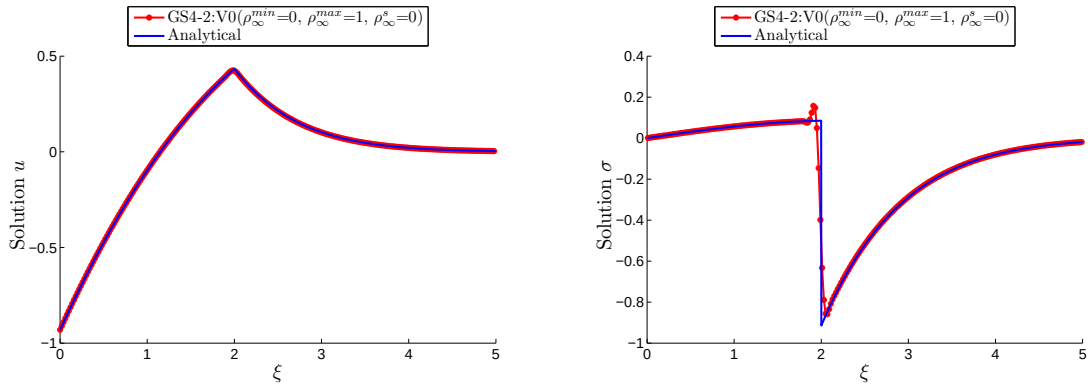
This problem serves to provide additional illustrations of the previous observations. First, it is worthy to note that this problem can be solved by the *i*Integrator in the same manner as done for the previous numerical example. Figures 7.9 to 7.12 show the plots of θ and $\dot{\theta}$ as a function of time for the specific node at $\xi = 0.02$ employing the two cases, i.e., the GS4-1 framework with and without selective control feature for ρ_∞ values of 1, 0.9, 0.8, and 0.7, using $\Delta\tau = 0.1$. It can be seen from these figures that the two cases yield good results of θ. However, for the solutions of $\dot{\theta}$, the case

[6]Note that in these figures, only the numerical solutions for the spatial domain $\xi \in [0, 5]$ are shown to clearly highlight the results at the point of discontinuity (i.e., $\xi = 2$)



(a) (b)

Figure 7.7: Plot of u and σ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of the spatial coordinate ξ at $\tau = 2$ employing GS4-2: V0($\rho_\infty^{min} = 0, \rho_\infty^{max} = 1, \rho_\infty^s = 0$) i.e., the U0-V0$_{optimal}$ with $\Delta\tau = 0.1$



(a) (b)

Figure 7.8: Plot of u and σ (for the Danilovskaya's First Problem, Section 7.5.4.1) as a function of the spatial coordinate ξ at $\tau = 2$ employing GS4-2: V0($\rho_\infty^{min} = 0, \rho_\infty^{max} = 1, \rho_\infty^s = 0$) i.e., the U0-V0$_{optimal}$ with $\Delta\tau = 0.01$

without the selective control feature results in large numerical oscillations. These oscillations can easily be reduced when the selective control feature is enabled (i.e., by choosing $\rho_\infty^s = 0$). This consistently shows the advantage of the GS4-1 framework with the selective control feature in contrast to the existing method without such a feature. Additionally, the numerical results of the dimensionless displacement (u) and stress (σ) as a function of the spatial coordinate ξ at $\tau = 2$ are shown in Figures

7.13 and 7.14 for $\Delta\tau$ values of 0.1 and 0.01, respectively. From these figures, it can be seen that the numerical solutions of these variables employing the U0-V0$_{optimal}$ are satisfactory.

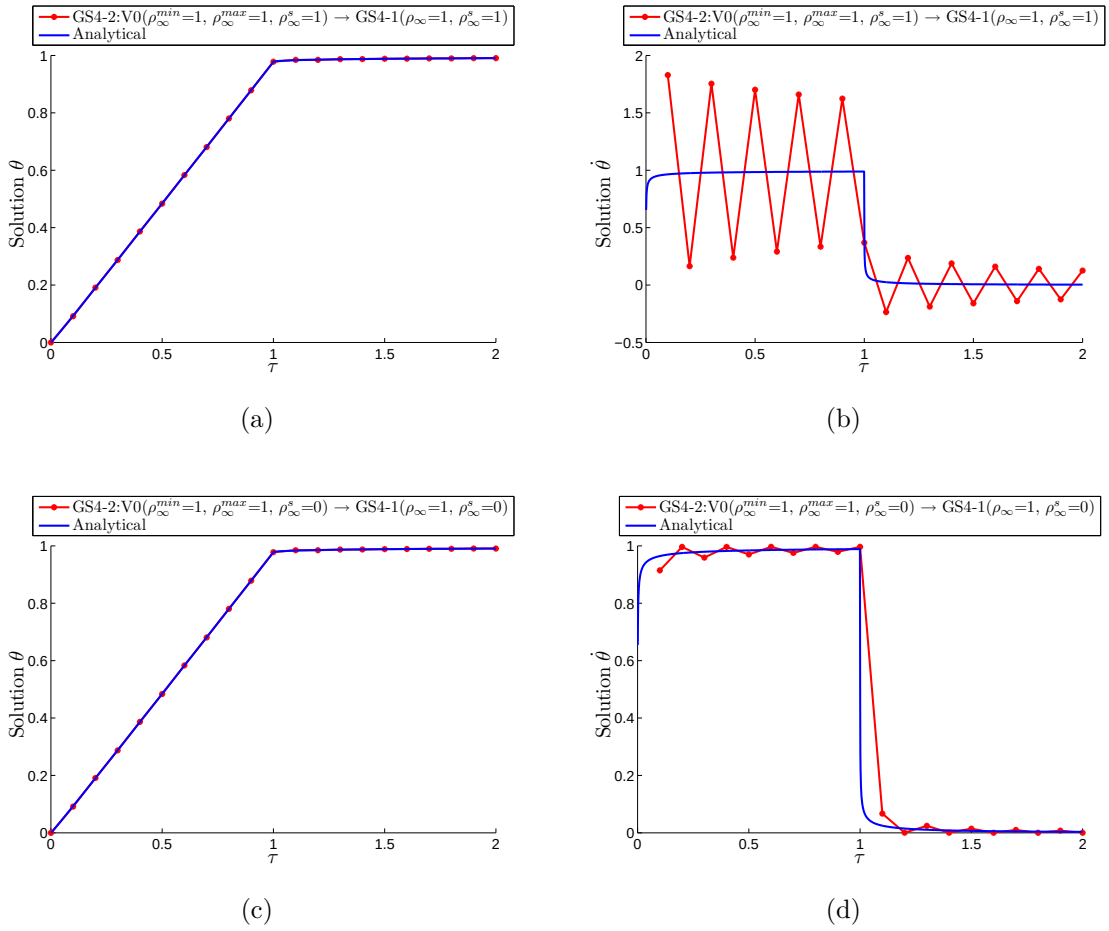


Figure 7.9: Plot of θ and $\dot{\theta}$ (for the Sternberg-Chakravorty's Problem, Section 7.5.4.2) as a function of time for specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 1$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 1, \rho_\infty^s = 0$) i.e., the case with selective control feature

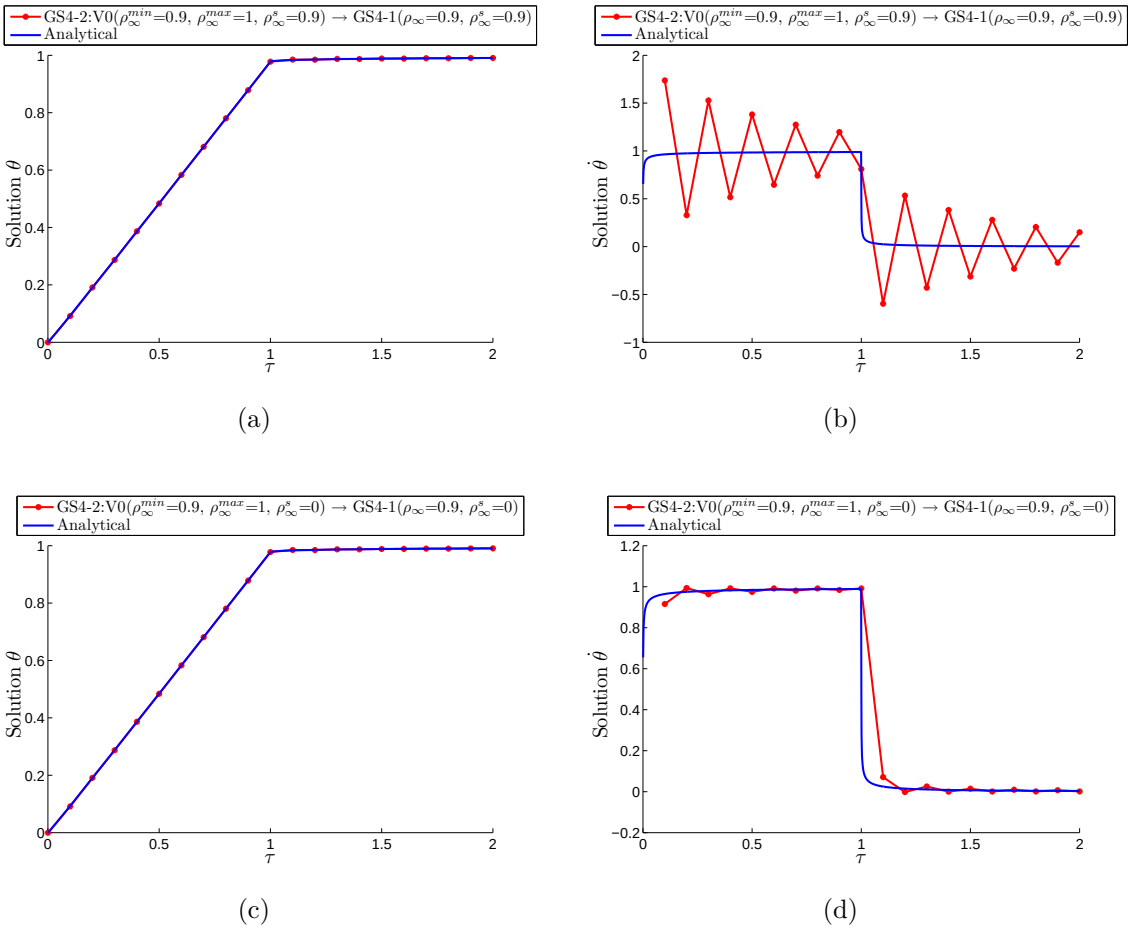


Figure 7.10: Plot of θ and $\dot{\theta}$ (for the Sternberg-Chakravorty's Problem, Section 7.5.4.2) as a function of time for specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 0.9$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 0.9, \rho_\infty^s = 0$) i.e., the case with selective control feature

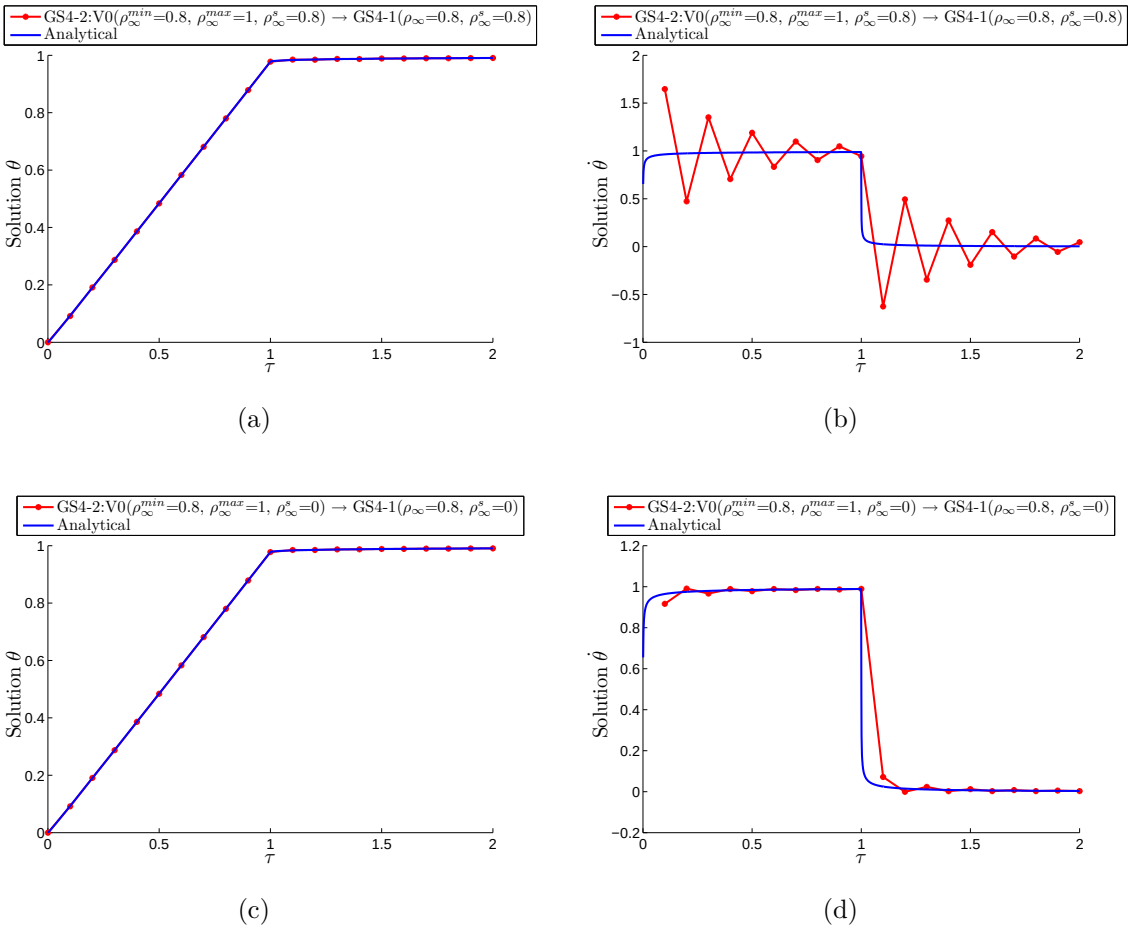


Figure 7.11: Plot of θ and $\dot{\theta}$ (for the Sternberg-Chakravorty's Problem, Section 7.5.4.2) as a function of time for specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 0.8$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 0.8, \rho_\infty^s = 0$) i.e., the case with selective control feature

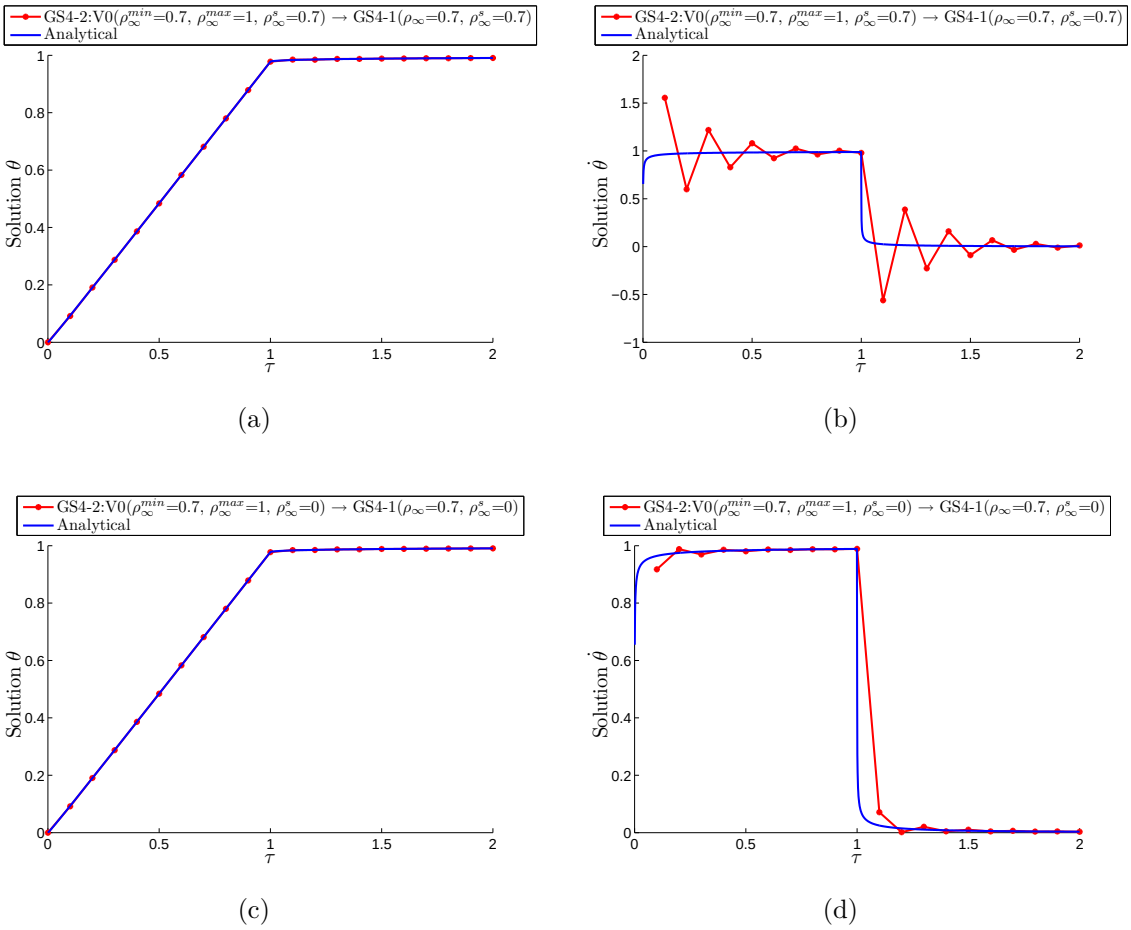
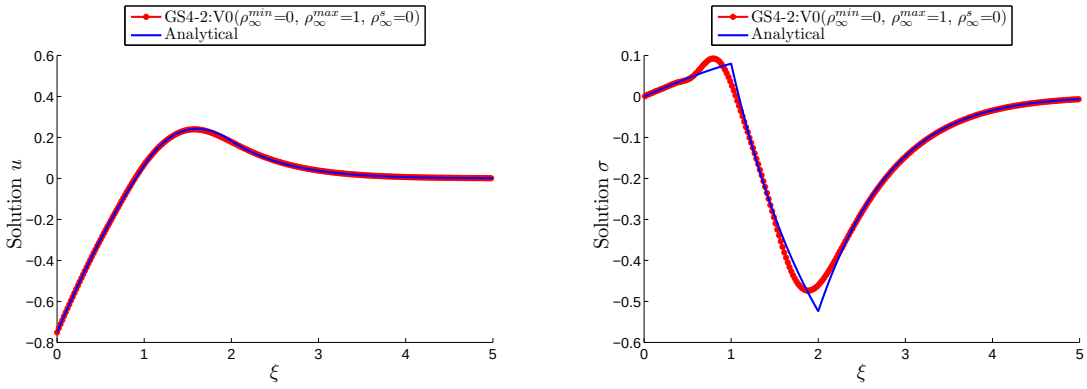
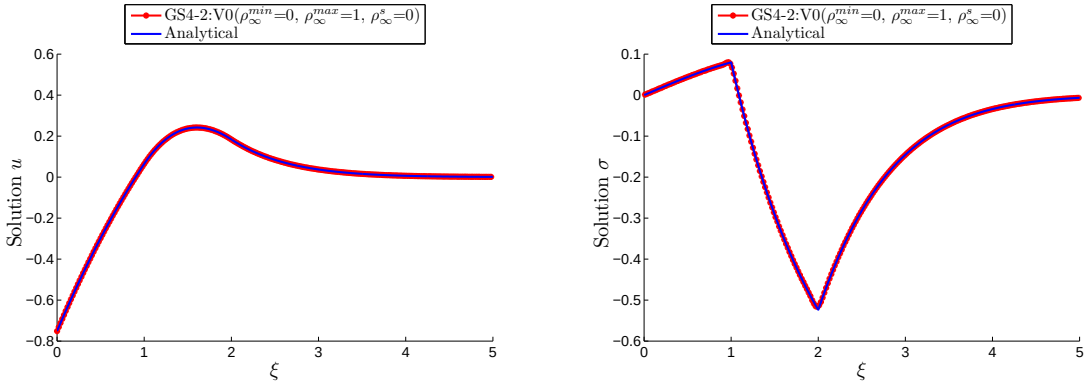


Figure 7.12: Plot of θ and $\dot{\theta}$ (for the Sternberg-Chakravorty's Problem, Section 7.5.4.2) as a function of time for specific node at $\xi = 0.02$ employing: (i) GS4-1($\rho_\infty = \rho_\infty^s = 0.7$) i.e., the case without selective control feature, and (ii) GS4-1($\rho_\infty = 0.7, \rho_\infty^s = 0$) i.e., the case with selective control feature



(a) (b)

Figure 7.13: Plot of u and σ (for the Sternberg-Chakravorty's Problem, Section 7.5.4.2) as a function of the spatial coordinate ξ at $\tau = 2$ employing GS4-2: V0($\rho_\infty^{min} = 0, \rho_\infty^{max} = 1, \rho_\infty^s = 0$) i.e., the U0-V0$_{optimal}$ with $\Delta\tau = 0.1$



(a) (b)

Figure 7.14: Plot of u and σ (for the Sternberg-Chakravorty's Problem, Section 7.5.4.2) as a function of the spatial coordinate ξ at $\tau = 2$ employing GS4-2: V0($\rho_\infty^{min} = 0, \rho_\infty^{max} = 1, \rho_\infty^s = 0$) i.e., the U0-V0$_{optimal}$ with $\Delta\tau = 0.01$

Chapter 8

Application of GS4-1 Framework to Silica Particle Deposition

As previously mentioned in Chapter 1, *Introduction*, part of the research in this study is to conduct a preliminary investigation on silica scaling, which is the practical problem of interest that is costly yet common in the geothermal power plants. Therefore, in this chapter, we turn our attention to the application of the newly developed GS4-1 framework to silica scaling, in particular that is due to the deposition of silica particles (i.e., silica in particulate/colloidal form), hence termed silica particle deposition. Unlike previous chapters, the discussion in this chapter is focused on the physics of the problem with the objective to gain more insights into the problem, and in particular into the effects of the flow and physicochemical parameters on the rate of deposition, so that more effective measure can be further developed from this study for the mitigatation of the problem. Of worth mentioning is that this problem is governed by a partial differential equation which is first order in time and therefore suits the purpose of the development of the GS4-1 framework presented in this thesis.

8.1 Introduction

Silica scaling is a major problem in the geothermal power industry which provides about 13% of New Zealand's total power generation [86]. It is particularly a problem in heat exchangers, re-injection pipelines, well casings and re-injection formation rocks of the power plant. The problem is a complex phenomenon involving

multidisciplinary areas and several strongly coupled mechanisms such as chemistry, heat transfer, and hydrodynamics. The problem occurs in geothermal power plants due to the presence of large quantities of quartz (SiO_2) in the permeable rocks of the geothermal reservoir. This quartz dissolves into the hot geothermal fluid, source of thermal energy subsequently converted in the power generation process, until it reaches an equilibrium concentration. As the geothermal fluid is brought to the surface from the hot geothermal reservoir, the pressure and temperature of the fluid decreases and the fluid becomes two-phase; steam and liquid. At the separator in the power plant, steam is separated from the geothermal fluid to provide sources for the turbine in the steam cycle, while leaving behind the liquid phase, often called the brine. Due to this process and the decrease in pressure and temperature, the quartz may then become super-saturated in the brine. At this point, colloidal particles of amorphous silica are formed instead of crystalline quartz. As they pass through the mild steel pipe, these colloidal particles are deposited to the walls forming deposit layers that eventually reduces the efficiency of the power generation. How this deposition occurs, how to model such process, and how the various flow and physicochemical parameters affect such process is the focus of the present study and will be discussed in this chapter. Another type of deposition due to the monomeric amorphous silica is also possible, but is beyond the scope of this thesis.

The costs incurred by silica deposition are significant. As large quantities of the brine are passed through the power station each day, the deposit build-up increases at a growth rates that can be quite significant, potentially blocking up the entire pipeline. Due to the presence of the deposit layers, greater pumping pressure is required to re-inject the geothermal fluid back to the reinjection well. Additionally, there is a large maintenance cost involving mechanical clean, chemical clean, and process clean. The most significant cost is due to the replacement of the re-injection well which can cost up to 10 million dollars [87]. The problem also results in loss of power generation during downturn time. Due to the large costs imposed by the silica deposition, the problem has gained great attention, both in research and technical implementations. The geothermal power station developers in New Zealand as well as around the world have been investigating methods for controlling the problem for decades. The most common methods used for such a purpose are:

1. Processing the brine at temperatures that prevent silica saturation.
2. Diluting the brine with fresh water to prevent silica saturation.
3. Reducing the pH of the brine to slow the polymerisation kinetics of silica.
4. Treating the brine with agents to prevent reaction of silica deposition co-products.
5. Removing silica from brine using precipitants such as lime.
6. Controlled precipitation of silica to prevent monomeric and polymerising silica deposition.
7. Cooling or rapid thermal quenching of geothermal brine.
8. Treating brine with silica scale inhibitors and dispersants.

In New Zealand, silica scaling has traditionally been mitigated by method (1), which can be achieved by keeping the geothermal brine at a temperature that is high enough to prevent supersaturation to occur. Although this may provide a solution to the problem, it is a less effective measure as the useable heat is not fully utilised and therefore the power output is not maximal. Additionally, method (3) has also been used in New Zealand as another prevention measure by acid-dosing the geothermal fluid before it is passed through the power station. By doing this, the polymerisation process of the amorphous silica is slowed down to a time scale that is long enough for the brine to be re-injected without scaling occurring in the plant. However, the acidification may lead to corrosion in the pipework due to the reaction with the acid if not carefully controlled.

This study, which is co-funded by Mighty River Power, one of the major power companies in New Zealand that operate geothermal power stations, is to be one of several studies in a series of research projects with an overarching goal of gaining a better understanding of the silica scaling process to ultimately be able to mitigate the problem more effectively. The objective in the present study is to develop a numerical model of colloidal deposition, based on a well-founded theoretical framework, that can be used to gain more insights into the problem, and

in particular into the effects of the flow and physicochemical parameters on the rate of deposition, so that more effective measures can be developed from such a study. The scope of the numerical model is, however, on the colloidal deposition only, leaving out the deposition of monomeric amorphous silica as the two processes are completely different from both the hydrodynamic and physicochemical point of view. Additionally, the numerical model developed in this study considers the colloidal deposition from a laminar, isothermal suspension only, as the case of turbulent flow, although more representative of the actual problem occuring at the power plant, is more complicated both theoretically and numerically. This work can therefore be viewed as a preliminary study to a more complicated, holistic research and development that will be undertaken in the near future in this series of research projects.

This part of the thesis describes the theory of particle deposition and the numerical modelling of such a process, which, upon validation using available experimental and numerical results from the literature, is employed to investigate the effect of the flow and physicochemical parameters involved in the deposition problem. The outline of the rest of the chapter is as follows. In Section 8.2, the literature on modelling silica colloid deposition in geothermal energy utilization as well as on modelling of colloidal deposition in general setting is first reviewed. Section 8.3 presents the theory of time dependent particle deposition problem including the governing equation, hydrodynamic interactions and colloidal interactions. In Section 8.4, the numerical model of particle deposition through parallel plate channel is presented. This is followed by the description of the computational procedure to perform the spatial discretization of the numerical model using the Finite Element Method (FEM) in Section 8.5.1. As previously mentioned, this first step in solving the time dependent problems numerically or computationally transforms the continuous partial differential equation to a set of ordinary differential equations that is easier to solve. The description of boundary conditions used in the numerical model is also presented. This is followed by the description of how the developed GS4-1 framework can be employed to solve the resulting system of ordinary differential equations in Section 8.5.2. In Section 8.6, we provide validation data of the particle deposition model developed in this study and solved using the described computational framework (i.e., FEM for spatial discretization and GS4-1 framework for time discretization). Having validated the

model, we then show in Section 8.7 the effect of the selective control feature of the GS4-1 framework for this problem to demonstrate the consistency of the performance. The validated model is then employed to investigate the effects of the various flow and physicochemical parameters on the rate of the deposition (i.e., parametric studies), and the results of the parametric studies are discussed in Section 8.8. Finally, we provide recommendations to mitigate the deposition problem based on the results of the parametric studies undertaken in this work. Note that some recommendations for future work in this subject matter are given in Chapter 9, *Concluding Remarks and Future Directions*.

8.2 Literature Review

8.2.1 Previous Work on Modeling Silica Particle Deposition in Geothermal Energy Utilization

Among the early work in modeling geothermal scaling is the work by Schroeder [88] where a simple model for modeling temperature-dependent scale deposition from geothermal brine was developed using a simple heat-transfer model with implicit formulation that is solved directly using matrix inverse methods appropriate for banded matrices. In this work, the problem is treated as a moving boundary problem. However, the hydrodynamic effects on the scale deposition are not taken into account which reduces the actual phenomenon to a simple static heat diffusion problem. Results show that the total deposition strongly depends on the deposition curve used in the model. Although temperature plays an important role in silica deposition from geothermal brines, it is important to include the hydrodynamic effects on the scale deposition model since the geothermal brine in which the silica colloids are suspended is not static, but rather is flowing, often in turbulent state. Therefore, hydrodynamic effects play an important role on the deposition mechanism and has been linked to the formation of ripples on the deposit surface as evidence in a study by Gudmundsson and Bott [89] who run fouling trials at two Icelandic fields, Hveragerdi and Svartsengi. Although the time-dependent scale deposition model developed by Schroeder may have the importance of representing initial modeling work on geothermal scaling, it is far from presenting the detailed phenomenological description of the scale deposition mechanism.

In another study, Jamieson [90] developed a model based on general logic to evaluate what happens to silica colloids as they flow along the length of a tube/duct. The aim was to predict silica deposition rate in tubes, noncircular ducts, or open drains for comparisons with actual field results. In this model, the transport of silica colloids from bulk suspension onto the deposit walls were represented by four transport processes, namely turbulent diffusion to the sublayer, inertial projection across sublayer, convection across sublayer by turbulent burst, and Brownian diffusion in the sublayer. Subsequently, the mass transfer coefficients for all these processes were combined to give an effective overall mass transfer coefficient for the deposition. Free flight models were used to determine the mass transfer coefficient for the inertial projection and convection across sublayer. These models are among the earliest attempts in modeling the deposition mechanism in an Eulerian framework and date back to 1957 with the Friedlander and Johnstone's landmark paper [91]. Although these models have contributed to the understanding of the deposition process, they have fundamental difficulties and are physically not satisfactory [92] including their piecemeal nature and the required empirical tuning, which hinder extrapolation to two- or three-dimensional flows with great confidence. In determining the effective overall mass transfer coefficient corresponding to the four distinct transport processes, Jamieson used a network model analogous to electrical circuit with four different resistances. Although this work provides an early exposition of computational model to predict silica deposition rate in geothermal applications, it lacks theoretical justification of the general logic used to represent the physical processes.

A model to determine the amount of deposited material due to scaling on geothermal heat exchanger has also been developed by Fryer [93] following a general fouling model developed initially in [94] based on the ideas of deposition and removal balance by Kern and Seaton [95]. However, such an analytical model lacks physical interpretation not only of the mechanisms responsible for both the deposition and removal of the deposited materials, but also the hydrodynamic effects on these processes. Motivated by the lack of understanding of the hydrodynamic effects on the silica deposition process, Pott and co-workers [96] derived equations leading to a numerical simulation of the initial deposition of silica colloids onto a flat plate in

laminar parallel flow. The objective was to investigate hydrodynamic effects on silica deposition process by numerically simulating the process using a computational fluid dynamic software package called PHEONICS with additional coding named GENTRA, to take into account the forces involved in transporting the colloids from the bulk fluid to the deposit surface. The continuous phase of the fluid is solved within the main module of PHOENICS using equations of fluid motion including conservation laws for mass and momentum, while the motion of the silica colloids representing the discrete phase is solved by a particle tracking approach in the subroutine GENTRA. The particle individual trajectory is solved using the particle position and momentum equations. In this work, concentration force, as given by van de Ven [97], is assumed to be responsible for the initial scaling of silica colloids. Following this, an important assumption used in this work is the form of the velocity and concentration profiles needed to determine the concentration force. These profiles, which take power series, were derived using a boundary layer concept. However, this power series assumption was not justified, hence its appropriateness is ambiguous.

Motivated by the need to develop nondestructive and noninterfering magnetic water treatment as a technique for scale prevention and removal, Chan and Moussa [98] developed a model using Lagrangian analysis to predict the trajectories and deposition of silica colloids onto two cylindrical probes under the influence of nonuniform magnetic field force and other relevant surface forces including London-van der Waals forces, viscous force, and added mass force. The developed model was used to explain findings from experimental investigation of magnetic effects on silica scale deposition on cylindrical surface in a crossflow. Although the predictions using the developed model were in qualitative agreement with experimental results, its practicality in terms of computational cost is ambigous since the model was developed in Lagrangian approach, which is known to be a good research tool for understanding the flow physics but is unlikely to be used as a design tool or a practical method for engineering calculations due to its intensive and expensive computations [92] as compared to the Eulerian approach. In this regard, it is profitable to develop the deposition model in the Eulerian framework since all practical CFD computations for a single-phase fluid in complex geometries are performed in this way, thus allowing easy integration of the model with established CFD codes.

The overview described above indicates that the previous modeling works on the deposition of silica colloids in geothermal utilization lacks theoretical basis and physical explanation of the various mechanisms involved in the deposition process. They also fail to accurately account for the hydrodynamic effects despite its importance. Therefore, there remains a need to develop a deposition model that not only takes into account these various mechanisms and hydrodynamic effects, but also is described in a consistent theoretical setting based on the conservation laws of physics that will naturally provide correct physical interpretations for a better understanding of the deposition phenomenon as well as the effects of the various flow and physicochemical parameters on the rate of the deposition. To this end, an overview of previous modeling works on colloidal deposition in a general theoretical setting is described next.

8.2.2 Previous Work on Modeling Particle Deposition in General

Understanding the mechanism of how particles of colloidal size ($1nm - 1\mu m$), suspended in a flowing liquid, deposit on walls, is not only of significant importance in geothermal energy utilization, but also of widespread interest in various situations/applications including manufacturing and process industries, natural aquatic environments, and medical sector. Research in this field has been focused on developing numerical models from which the deposition kinetics and the effects of the various flow and physicochemical parameters can be used to predict and control the deposition rate. The real problems of colloidal deposition, such as the silica colloids deposition onto heat exchangers, pipelines and reinjection wells in geothermal power plant, are often very complex due to the nature of such real problems. For instance, the colloidal particles may not be of uniform spherical shape, the suspension may be polydispersed, the wall may be far from being a smooth surface, and the physicochemical properties of the particles and wall such as the surface charge may be heterogenous in nature. These nonuniformities among others result in systems that are very difficult if not impossible to describe in sufficient detail to enable a complete theoretical/numerical analysis to be undertaken. Therefore, model studies inevitably have to consider 'ideal' systems defined by: (1) a number of simplifying assumptions, such as uniform spherical particles, smooth wall surfaces, constant physicochemical

properties, and dilute suspension in which case the interactions between the colloidal particles are negligible and only the interaction between the colloidal particles and the wall surface needs to be considered, (2) simpler/ideal and well-defined collector geometry such as parallel plate channels, rotating disc system, stagnation-point flow, and spherical collector, and (3) laminar flow, for which analytical solutions of the flow fields of the carrying fluid are available, reducing the degree of complexity of an already complicated problem. Ideal collector configurations such as parallel-plate and cylindrical channels have been extensively used in colloidal deposition studies [99–103] for a long time due to its significant importance. Although these ideal systems may not be fully representative of the real problems, such model studies may provide useful and valuable information leading to a better understanding of the real situations. Furthermore, studies of colloidal deposition involving experimental work often use such ideal systems due to their well-defined nature. Therefore, comparisons of the theoretical and experimental results can easily be undertaken to validate the accuracy of the numerical model and theoretical findings. The validated numerical model can then be used to investigate the effects of the various flow and physicochemical parameters on the rate of the deposition. From these results, a qualitative physical interpretation of the real situations can at least be made.

Earlier theoretical works in modeling colloidal deposition ([104, 105] among others), took into account various external forces such as gravity and electrostatic forces in the model but were unable to capture the deposition kinetics due to the absence of surface/colloidal interactions in the developed models. It has since then been recognized that these surface-colloid interactions play an important role in adsorpting the colloidal particles onto the surface of the deposit wall once the colloids arrive in the vicinity of the wall. Because of the significance of the interactions, numerous research works to quantify them have been undertaken and published in the literature, pioneered by well-known DLVO theory [106]. According to this theory, the total colloidal interaction between the colloids and deposit wall, which leads to the adsorption of the colloids onto the surface of the wall, is the sum of a dispersion force, known as the van der Waals interaction [107–109] and the electrical double layer interaction. The latter can lead to significantly different deposition behaviours, depending on the surface charge of both the particles and the wall surface. When the

two have similar charge, the electrical double layer potential is repulsive in nature, giving rise to an energy barrier that hinders the attachment of the particles to the wall surface, reducing the deposition rate. In such case, the deposition problem is often said to be under unfavorable chemical condition. On the other hand, the electrical double layer is attractive in nature when the particles and wall surface are oppositely charged, in which case the deposition rate is enhanced and the problem is usually referred to as being under favorable chemical condition. The reviews and detailed discussions of the DLVO theory are available in [97, 100, 106] among others. In particular, the applications of the theory to colloidal deposition problems can be found in [100].

The particle deposition problem is, in general, governed by a convection-diffusion equation, which describes the concentration and transport of the suspended particles due to: (1) convection by the carrying fluid flow field, (2) DLVO interactions and other external forces involved such as gravity and magnetic force, (3) Brownian diffusion resulting from the colloidal size of the particles, and (4) effect of any sink or source term if available. The particle deposition rate can be determined by solving such equation, mostly numerically due to the presence of the DLVO interactions and hydrodynamic effects which hinder the derivation of analytical solutions. However, under certain/simplified physicochemical conditions, analytical approximations for the deposition rate can be derived such as the Smoluchowski-Levich approximations [99, 110] for the particle deposition problems under favorable chemical conditions (i.e., when the electrical double layer is attractive). This approximation simplifies the complex problem by assuming that the hydrodynamic interaction between the particles and the wall surface is counterbalanced by the van der Waals attraction between the two surfaces, and that all other colloidal interactions and external forces are absent/negligible. With such simplifying assumptions and under steady state condition, analytical solutions to the resulting governing equation can be derived and are available in the literature for some well-defined/ideal geometries, such as rotating disc or stagnation-point flow collector [99], plate in a uniform flow [99], cylinder in a uniform flow [99], parallel-plate or cylindical channels [100–102], and moving wire or plate [103]. These analytical solutions can be used for testing the accuracy of a numerical solution. In solving the governing equation numerically, appropriate

boundary condition at the wall surface must be used. Due to the limitation in the present understanding about the hydrodynamics and physicochemical conditions near the wall surface, researchers in this field have made various simplifying assumptions about the wall boundary condition to allow for a complete description of the numerical model of the particle deposition. One of the commonly used boundary conditions at the wall surface is the so-called perfect-sink model, which assumes that all particles arriving at or close enough to the wall surface will be irreversibly adsorbed immediately and subsequently disappear from the sytem. Such simplified wall boundary condition model is appropriate for use in particle deposition problem under favorable chemical conditions (i.e., when the electrical double layer is attractive). On the other hand, for the deposition case under the unfavorable chemical condition (i.e., when the electrical double layer is repulsive), other types of wall boundary condition have been proposed including the constant migration flux [111] and non-penetration model [112]. Numerical procedure for solving the complete numerical model was presented in [113] including the boundary condition, numerical discretization, mesh refinement, operational criteria for numerical stability, and allocation of the mesh grids. The developed models allow for useful studies of the principal mechanisms involved in the deposition process and are useful for investigating the effects of the various flow and physicochemical parameters, such as ionic strength, particle size, surface potentials, Hamaker constant, double layer interaction mode, and fluid velocity, on the rate of the deposition [113–115].

For particle deposition problem with repulsive electrical double layer potential, resulting from the particle and wall surfaces being similarly charged, and leading to the presence of an energy barrier that reduces the deposition rate, approximation model is also available such as the Surface/ Interaction Force Boundary Layer Approximation (SFBLA or IFBLA) model. This model was developed and applied originally by Ruckenstein and Prieve [116] and Spielman and Friedlander [117] for spherical and cylindrical collectors, and later was extended to parallel plate and cylindrical channels by Bowen and co-workers [118]. The simplifying assumption underlying the model is that the surface/colloidal forces act only in a very small region in the vicinity of the wall surface as compared to the region/thickness of diffusion boundary layer. This allows for the decoupling of the convection effect

due to the carrying fluid flow field and the surface/colloidal force effects, hence simplifying the governing equation. Subsequently, the mathematical formulation of the model leads to the incorporation of the surface force effects as a wall boundary condition, which takes the form of a first order reaction, with the reaction rate constant representing the surface force effects. In this way, the model was hoped to tackle the challenge of numerically solving the particle deposition problem when repulsive electrical double layer potential is present. Although the model was claimed to revert to the favorable deposition case (i.e., when the particle and wall surfaces are oppositely charged) by the choice of infinite reaction rate constant and yield good agreement with experimental results, the consistency of such an argument is ambigious. As shown previously by Adamczyk and co-workers [119, 120], the reaction rate constant for such a case takes a negative value instead of infinite, losing its physical interpretation [102] and consistency of the arguments. Moreover, the model neglects the effect of a secondary energy minimim [102] despite possible accumulation of particles in this region [102].

Since the SFBLA approximation cannot be used to model particle deposition problem when there exists coupling between the electrical double layer interaction and the fluid convection, Song and Elimelech [111] proposed a sophisticated numerical procedure employing additional operational criteria for a stable numerical solution of such deposition problem. A new boundary condition, called the constant migration flux was also proposed, claiming that the classical Perfect-Sink boundary condition is not suitable for the numerical model of particle deposition when repulsive colloidal interactions are present. A semi-empirical approach has also been developed based on dimensional analysis [121] as an alternative approach to predict particle deposition rate in terms of collision efficiency for unfavorable deposition with particular application in porous media. Although the SFBLA model was developed for the unfavorable deposition case, large discrepancies between the numerical solutions and the experimental results were observed for such cases [122]. Additionally, the combination of a numerical method and SFBLA approximation has been used by Sjollema and Busscher [9] to solve particle deposition in a parallel plate channel. In contrast to the original SFBLA model by Ruckenstein and Prieve [116], Spielman and Friedlander [117], and Bowen and co-workers [118], the model by Sjollema and

Busscher [9] yields satisfactory results that agree well with experimental data for most of the experimental cases.

It is worthy to note that the previous efforts in developing a numerical model of particle deposition as described above consider only the initial stage of the deposition process, in which case steady state has been commonly and appropriately assumed. The results from such efforts can be used as basic studies to gain an understanding of the particle deposition problem such as the interactions at the interface between the colloidal particles and the wall surface, and the effects of the various flow and physicochemical parameters, such as ionic strength, particle size, surface potentials, Hamaker constant, double layer interaction mode, and fluid velocity, on the rate of the deposition. However, the initial deposition stage, which is linear and steady state in nature, is usually short-lasting. The real deposition problem, as we know it, is a dynamic process, as evident by the ripple-shape pattern of typical deposition surface, resulting from the accumulation of particles at the particle/wall interface during the deposition period. Such dynamic process is, however, very complex which prohibits any exact mathematical treatment or analytical solutions to the problem. In this regard, numerous efforts in the past to model and capture the dynamic part of deposition problem usually involve drastically simplified models such as the Langmuirian and the Random Sequential Adsorption (RSA) approaches (see Reference [123–126] among others), which rely on the use of stochastic methods such as Monte-Carlo simulation technique. In this work, however, we wish to view the particle deposition problem as a dynamic process from a consistent mathematical setting and physics of continuum mechanics, without having to resort to any stochastic methods. For this purpose, the deposition problem is expressed in its general form as a time-dependent (first order) problem involving convection and diffusion transports. We next describe the theory and computational procedure to solve such a problem.

8.3 Theory of Particle Deposition

8.3.1 Governing Equations

We shall consider from the start the particular case of particle deposition in laminar, isothermal, incompressible flow, through ideal collector for the reasons cited above. For this purpose, we choose a parallel plate channel of depth $2b$ as shown in Figure 8.1 through which a dilute suspension of spherical particles is flowing. The suspension is further assumed to be stable such that all interactions between the colloidal particles can be neglected. Let us choose a fixed Cartesian coordinate system (x, y, z) having its origin $(x = 0)$ at a point sufficiently far from the entrance region that the carrying fluid velocity can be assumed as being fully developed. At this point, the concentration of the particles must be specified (i.e., quasi-initial condition). It is futher assumed that no deposition occurs prior to this point. The general continuity equation, which governs the transport and deposition of the particles, under such conditions can be written as follows

$$\frac{\partial n}{\partial t} + \nabla \cdot \vec{J} = Q \tag{8.3.1}$$

where n is the particle number concentration, Q is a sink/source term, and $\vec{J} = (J_x, J_y, J_z)$ is the particle flux given by

$$\begin{aligned} J_x &= -D_{\|}\frac{\partial n}{\partial x} + U_x n, \\ J_y &= -D_{\|}\frac{\partial n}{\partial y} + U_y n, \\ J_z &= -D_{\perp}\frac{\partial n}{\partial z} + U_z n, \end{aligned} \tag{8.3.2}$$

where (J_x, J_y, J_z) are the respective components of the particle flux vector $(\vec{J})$, $D_{||}$, $D_{\perp}$ are the tangential and normal components of the Brownian diffusion tensor, and U_x, U_y, U_z are the components of the particle velocity vector due to all forces and torques acting on the particles. These components of the velocity vector can be expressed as follows [102]

$$
\begin{aligned}
U_x &= \frac{1}{\mu} K_{r,\|} (K_{t,\|} K_{r,\|} - K_{c,\|}^2)^{-1} \left(-\frac{\partial \phi}{\partial x} \right) + F_h(z) V_x, \\
U_y &= \frac{1}{\mu} K_{r,\|} (K_{t,\|} K_{r,\|} - K_{c,\|}^2)^{-1} \left(-\frac{\partial \phi}{\partial y} \right), \\
U_z &= \frac{1}{\mu} K_{t,\perp}^{-1} \left(-\frac{\partial \phi}{\partial z} \right),
\end{aligned}
\tag{8.3.3}
$$

where μ is the dynamic viscosity of the carrying fluid, $K_{t,\|}$, $K_{t,\perp}$ are the tangential and normal components of the particle translational resistance dyadic, $K_{c,\|}$ is the tangential component of the coupling dyadic, $K_{r,\|}$ is the tangential component of the rotation dyadic, ϕ is the total interaction potential acting on the particles, which include the external and wall-particle colloidal potential, $V_x = \frac{3}{2} V_m \frac{z}{b} \left(2 - \frac{z}{b}\right)$ is the $x-$component of the fluid velocity vector (which, in this particular case, equals the magnitude of the fluid velocity itself, $|\mathbf{V}|$), V_m is the mean/average velocity, and $F_h(z)$ is a hydrodynamic correction function for the deviation of the particle velocity from the carrying fluid velocity resulting from hydrodynamic disturbance due to the presence of the bounding walls. In equation (8.3.3), external torques on the particles are assumed small and thus neglected. Furthermore, in the theoretical analyses of particle deposition problem under the above circumstances, it is commonly assumed that the tangential component of the Brownian diffusion tensor $D_{||}$ is neglegible [106].

Assuming that the width of the channel is much larger than the diffusion boundary layer thickness in which case edge effects are negligible and that the y component of the external force is independent of the y coordinate, Equation (8.3.1) to (8.3.3) can be reduced to a two-dimensional form. However, it is still very inconvenient to solve the resulting two-dimensional partial differential equation using numerical methods because of the great number of physical variables involved in the problem. Therefore, a further simplification is necessary and this can be achieved by transforming the equation into dimensionless form by use of dimensionless variables defined as

$$
H = \frac{z}{a} - 1, \quad \bar{x} = \frac{x}{b}, \quad c = \frac{n}{n_\infty}, \quad A = \frac{a}{b}, \quad Pe = \frac{3 V_m a^3}{2 b^2 D_\infty}, \quad \bar{\phi} = \frac{\phi}{kT}, \quad t = \frac{t\, D_\infty}{a^2} \tag{8.3.4}
$$

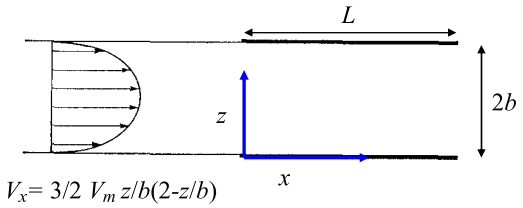


Figure 8.1: Parallel plate channel configuration. Flow is laminar, assumed fully developed. Particle deposition occurs at $x \geq 0$

where H is the dimensionless separation distance between the surfaces of the wall and the particle, c the dimensionless particle concentration, n_∞ the particle number concentration in the bulk at the origin (inlet), Pe the Peclet number, D_∞ the particle diffusion coefficient in the bulk, $\bar{\phi}$ the dimensionless potential, k the Boltzmann constant, and T the absolute temperature (in K). Substituting these dimensionless variables into the partial differential equation yields the two-dimensional governing equation in dimensionless form as follows (in the absence of sinks or sources, i.e., $Q = 0$)

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial \bar{x}}\left[-A^2 \bar{M}_{\parallel}\frac{\partial c}{\partial \bar{x}} + PeF_h(H)(H+1) \times [2-(H+1)A]c - \bar{M}_{\parallel}\frac{\partial \bar{\phi}}{\partial \bar{x}}Ac\right] + \frac{\partial}{\partial H}\left[\bar{K}_{t,\perp}^{-1}\left(-\frac{\partial c}{\partial H} - \frac{\partial \bar{\phi}}{\partial H}c\right)\right] = 0 \tag{8.3.5}$$

where $\bar{M}_{\parallel} = 6\pi a K_{r,\parallel}(K_{t,\parallel}K_{r,\parallel} - K_{c,\parallel}^2)^{-1}$ and $\bar{K}_{t,\perp} = K_{t,\perp}/6\pi a$. For typical experimental conditions, $A \ll 1$, and therefore $A^2 \to 0$ and can be neglected, and the coefficients $\bar{K}_{t,\perp}^{-1}$ and $F_h(H)$ can be approximated by the universal hydrodynamic correction functions $F_1(H)$ and $F_3(H)$, respectively. Making use of these simplifications, the dimensionless governing equation, equation (8.3.5), can be expressed in final form as follows

$$\frac{\partial c}{\partial t} + \frac{\partial c}{\partial \bar{x}}\left(PeF_3(H)(H+1)(2-(H+1)A)\right) - \frac{\partial}{\partial H}\left[F_1(H)\left(\frac{\partial c}{\partial H} + \frac{\partial \bar{\phi}}{\partial H}c\right)\right] = 0 \tag{8.3.6}$$

8.3.2 Hydrodynamic Interactions

In this section, we describe the theory of the hydrodynamic interactions that are relevant to the particle deposition described above. Note that such interactions have been included in the formulation of the governing equations described in the preceeding section. Here, the objective is to highlight briefly these interactions in a general sense.

When subjected to a flow, a freely moving particle follows the streamlines and the velocity of the particle $\vec{U}$ is equal to the local fluid velocity $\vec{V}$. However, when a rigid interface is present, the particle velocity differs from the local fluid velocity at the same point in the vicinity of the interface. The presence of the solid interface causes viscous resistance to the fluid motion. As a results, the particle lags the fluid flow due to such an increased hydrodynamic drag [8]. To account for this hydrodynamic interactions near the interface, the particle velocity needs to incorporate the correction to the particle mobility for the movement in the vicinity of the wall surface using the so-called hydrodynamic correction functions. These functions have been derived in [3–5, 7] for small spherical particles of a dilute suspension when in a slow streaming motion past a large obstacle. The derivations of these functions involve setting out the solutions of the Stokes equations satisfying the equation of continuity and non-slip condition on the interface, followed by the computation of the resulting mechanical action on the particles from which the hydrodynamic forces and consequently the hydrodynamic correction functions can be determined. Using these functions, the particle velocity in the vicinity of the interface can be expressed as follows [8, 106]

$$U_{\parallel} = F_3(H)V_{\parallel} \tag{8.3.7}$$

$$U_{\perp} = F_1(H)F_2(H)V_{\perp} \tag{8.3.8}$$

where $\parallel$ and $\perp$ indicate the parallel and normal/perpendicular components of the particle and carrying fluid velocity vectors. Note that in the particular particle deposition described above, there is only a parallel component of the carrying fluid velocity, i.e., $V_{\parallel} = V_x$ as illustrated in Figure 8.1. Meanwhile, the Brownian diffusion tensor $\bar{\bar{D}}$ in the bulk suspension is a scalar (denoted by D_∞), which is equivalent to the Brownian diffusion coefficient of a single particle of radius a given by

$$D_\infty = \frac{kT}{6\pi\mu a} \tag{8.3.9}$$

where μ is the dynamic viscosity of the carrying fluid. Similarly, in the vicinity of the wall surface, the diffusion tensor has to be adjusted to account for the reduction of particle mobility in such region using the hydrodynamic correction functions as follows

$$D_{\parallel} = F_4(H) D_\infty \tag{8.3.10}$$

$$D_{\perp} = F_1(H) D_\infty \tag{8.3.11}$$

where $\parallel$ and $\perp$ indicate the parallel and normal/perpendicular components of the particle Brownian diffusion tensor, respectively. In the above equations, $F_1(H), F_2(H), F_3(H)$, and $F_4(H)$ are the hydrodynamic correction functions originally derived and described in [3–5, 7]. Analytical expressions of these functions are given in [8] as follows

$$F_1(H) = \frac{19H^2 + 4H}{19H^2 + 26H + 4} \tag{8.3.12}$$

$$F_2(H) = 1 + \frac{1.79}{(0.828 + H)^{1.167}} \tag{8.3.13}$$

$$F_3(H) = \left\{ \begin{array}{ll} \frac{1}{0.754 - 0.256\ln(H)} & H < 0.137 \\ 1 - \frac{0.304}{(1+H)^3} & H \geq 0.137 \end{array} \right\} \tag{8.3.14}$$

$$F_4(H) = \left\{ \begin{array}{ll} \frac{1}{1.062 - 0.516\ln(H)} & H < 0.11 \\ \left(\frac{H}{2.639+H}\right)^{1/4} & H \geq 0.11 \end{array} \right\} \tag{8.3.15}$$

where H is the dimensionless separation distance between the particle and wall surface as given by equation (8.3.4). The values of these hydrodynamic correction functions are shown graphically in Figure 8.2. The above expressions describe the mobility of the particle close to a solid wall which take into account the hydrodynamic drag on the particle motion due to the viscous resistance to the fluid motion exerted by the solid interface. Figure 8.2 shows that the hydrodynamic interactions with the wall starts to influence the motion of the particle at separation distance $H < 3$ and the reduction in the mobility of the particle is greater as the particle moves closer towards the wall.

8.3.3 Colloidal Interactions

Understanding how the particles, suspended in the flowing fluid, deposit on the wall surface requires a detailed understanding of the solid-liquid interfacial region, particularly the interaction/colloidal potentials (ϕ) between these surfaces. These colloidal potentials consist of several components, the main ones being the electrical double layer and the van der Waals potential as described by the DLVO theory [127, 128, 106]. These principal potentials act over relatively short separation distance between the particle and wall surfaces, and are dependent on the surface properties of both the particles and the wall.

Following [9], the total colloidal potential is thought to be composed of the Lifshitz-van der Waals interaction ϕ_{LW} and the electrostatic double layer potential ϕ_{EDL} as

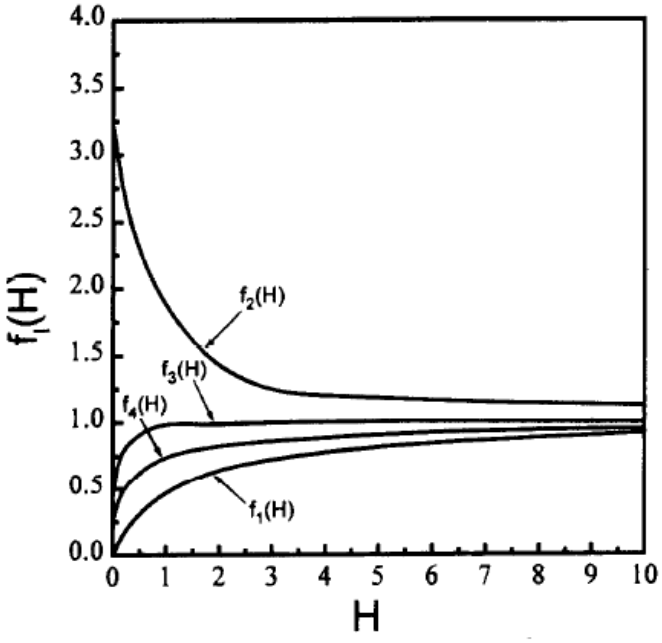


Figure 8.2: Universal hydrodynamic correction functions as a function of the dimensionless separation distance between a particle and a surface for a suspended particle in motion near an impermeable rigid wall [3–8]

described by DLVO theory, i.e.,

$$\phi = \phi_{LW} + \phi_{EDL} \tag{8.3.16}$$

$$\bar{\phi} = \frac{\phi}{kT} \tag{8.3.17}$$

where these potentials are defined as [9]

$$\phi_{LW} = \frac{-A_{132}}{6H}, \quad H \ll 1, \tag{8.3.18}$$

$$\phi_{EDL} = \pi\epsilon\epsilon_0 a(\zeta_1^2 + \zeta_2^2)\left[\frac{2\zeta_1\zeta_2}{\zeta_1^2 + \zeta_2^2}\ln\left(\frac{1+\exp^{-\tau H}}{1-\exp^{-\tau H}}\right) + \ln(1-\exp^{-2\tau H})\right] \tag{8.3.19}$$

In the above expressions for the potentials, A_{132} is the Hamaker constant for

interactions of particle 1 in medium 3 with a planar surface 2, ζ_1, ζ_2 are the electrical potential (i.e., the zeta potential at shear plane) of the particle and wall, ϵ is the relative permittivity or dielectric constant of the medium and ϵ_0 is the permittivity under vaccum, $\tau = \kappa a$ is the dimensionless Debye-Huckel reciprocal length, and $\kappa = \sqrt{\frac{2e^2 n_\infty z^2}{\epsilon kT}}$, where e is the electron charge, z is the valence, and $n_\infty = 1000 N_A C_s$ is the bulk number density of ions (ion number concentration), N_A is the Avogadros number, and C_s is the electrolyte molar concentration (ionic strength of the solution in Molar).

In addition, there is also a potential due to gravity and buoyancy ϕ_{GR}, defined as

$$\phi_{GR} = \frac{4}{3}\pi a^4 (\rho_p - \rho_l) g H \tag{8.3.20}$$

where ρ_p, ρ_l are the material densities of the particles and the carrying fluid, respectively.

8.4 The Numerical Model

The particle deposition problem, as described in Section 8.3, is not only complicated due to the multidisciplinary nature of the problem involving several strongly coupled mechanisms such as chemistry, heat transfer, and hydrodynamics, but is further complicated in the computational implementation aspects due to the nature of the colloidal potentials involved, namely the Lifshitz-van der Waals interaction ϕ_{LW} and the electrostatic double layer potential ϕ_{EDL}. These potentials, especially in the case of repulsive ϕ_{EDL} potential, have large gradients over a very narrow range (in the order of hundreds nanometer) along the direction perpendicular to the wall. From a computational implementation perspective, the nature of the potentials requires that the region where these potentials act is discretized in space using elements that are small enough to capture the gradients of the potentials. As a result, the number of elements required to represent such a very small region is tremendous. This mades the model computationally expensive, both in terms of memory capacity, but also in terms of the required running time. Because of this issue, it is not computational efficient to model the deposition problem by implementing the theory, as described

in Section 8.3, directly without enforcing some sort of simplification that enables the computation while still representing the complicated theory of the problem.

For this reason, in this work, we employed the numerical model of Sjollema and Busscher [9] which has been shown to provide good approximations to the experimental results of the deposition rate for some physicochemical conditions. This model follows the idea of Surface Force Boundary Layer Approximation (SFBLA) model originally developed by Ruckenstein and Prieve [116] and Spielman and Friedlander [117] and later extended to other application types by Bowen and co-workers [118]. The idea behind these models is to divide the flow channel into two regions (wall and core regions), from which the analytical solution to the wall region is used as the boundary condition to the core region that is to be solved numerically. However, unlike the SFBLA model which completely decouples the two regions (i.e., the convection term is neglected in the wall region, whereas the potential energy and hydrodynamic interaction terms are neglected in the core region), the model of Sjollema and Busscher considers solving the exact convection diffusion equation in the core region which includes the potential energy and hydrodynamic interaction terms. This way, the physics of the problem can be captured more correctly.

According to the model by Sjollema and Busscher, the governing equation (equation (8.3.6)) will be numerically solved for the core region ($H \gg \delta$) where δ is the dimensionless thickness of the diffusion boundary layer [9]. Meanwhile, for the wall region, $0 \leq H < \delta$, convective transport is neglected according to the SFBLA method, and the mass transport in this region toward the wall surface is governed by a diffusion equation which can be written, assuming steady state, as follows

$$-\frac{\partial}{\partial H}\left[F_1(H)\left(\frac{\partial c}{\partial H}+\frac{\partial \bar{\phi}}{\partial H}c\right)\right]=0 \tag{8.4.21}$$

It is assumed that the particle deposits on the wall as a perfect-sink, i.e., $c = 0$ at $h = 0$, while on the top boundary ($H = \delta$) we have $c = c_\delta$ which is a function of the dimensionless length $\bar{x}$. Using integrating factor, equation (8.4.21) together

with the boundary conditions can be solved analytically to yield the following initial dimensionless deposition rate $Sh_0(\bar{x})$ at $H = \delta$ as a function of the dimensionless length $\bar{x}$

$$\bar{J}_0(\bar{x}) = Sh_0(\bar{x}) = \left[F_1(H)\left(\frac{\partial c}{\partial H} + \frac{\partial \bar{\phi}}{\partial H}c\right)\right]_{|H=\delta} = \frac{c_\delta(\bar{x})}{\exp(-\bar{\phi}(\delta))\int_0^\delta \frac{\exp(\bar{\phi}(H))}{F_1(H)}dH} \tag{8.4.22}$$

In equation (8.4.22), $c_\delta(\bar{x})$ is the dimensionless particle concentration at $H = \delta$ (unknown) that is to be solved numerically. Equation (8.4.22) will be used as the boundary condition at $H = \delta$ for the core region, $\delta \leq H \leq \frac{b}{a} - 1$ governed by equation (8.3.6). This way, only the core region needs to be solved numerically while the colloidal interactions are taken into account using equation (8.4.22) as the boundary condition. Figure 8.3 shows the boundary and initial conditions for the numerical model of silica particle deposition employed in this study.

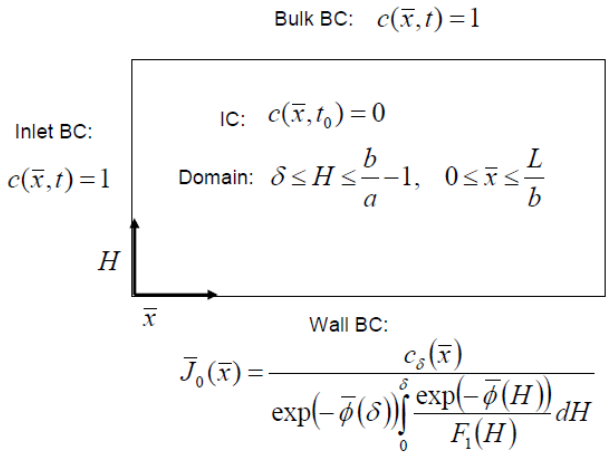


Figure 8.3: Boundary and initial conditions for the employed numerical model

8.5 Computational Procedures

8.5.1 Spatial Discretization

As previously mentioned, solving any time dependent problems such as the particle deposition problem governed by equation (8.3.6) numerically or computationally

usually involves two major steps. This section describes the first step, which is the spatial discretization. For this purpose, we choose the Finite Element Method for consistency with the rest of the thesis.

8.5.1.1 The Finite Element Method

In this study, we employ the Finite Element Method (FEM) to discretize the problem's spatial domain, and such procedure can be accomplished as follows.

Employing the method of weighted residuals to the dimensionless governing equation (equation (8.3.6)), for an arbitrary element (e) of r nodes bounded by $\Omega^{(e)}$ domain, requires that

$$\int_{\Omega^{(e)}} W_i^{(e)} \left\{ \frac{\partial c^{(e)}}{\partial t} + \frac{\partial c^{(e)}}{\partial \bar{x}} \left(PeF_3(H)(H+1)(2-(H+1)A) \right) \right\} \partial\Omega$$
$$- \int_{\Omega^{(e)}} W_i^{(e)} \left\{ \frac{\partial}{\partial H} \left[F_1(H) \left(\frac{\partial c^{(e)}}{\partial H} + \frac{\partial \bar{\phi}}{\partial H} c^{(e)} \right) \right] \right\} \partial\Omega = 0, \quad i = 1, 2, ..., r \tag{8.5.23}$$

where $W_i^{(e)}$ is the element weighting function. Integrating the diffusion term in equation (8.5.23) using Gauss theorem yields

$$\begin{aligned} &\int_{\Omega^{(e)}} W_i^{(e)} \frac{\partial}{\partial H} \left[F_1(H) \left(\frac{\partial c^{(e)}}{\partial H} + \frac{\partial \bar{\phi}}{\partial H} c^{(e)} \right) \right] \partial\Omega \\ &\quad = \oint_{\Gamma^{(e)}} W_i^{(e)} \left[F_1(H) \left(\frac{\partial c^{(e)}}{\partial H} + \frac{\partial \bar{\phi}}{\partial H} c^{(e)} \right) \right] \partial\Gamma \\ &\quad - \int_{\Omega^{(e)}} \frac{\partial W_i^{(e)}}{\partial H} \left[F_1(H) \left(\frac{\partial c^{(e)}}{\partial H} + \frac{\partial \bar{\phi}}{\partial H} c^{(e)} \right) \right] \partial\Omega \end{aligned} \tag{8.5.24}$$

where the boundary $\Gamma^{(e)}$ of the element is the surface at $H = \delta$ in the core region. Therefore, using the boundary condition according to the SFBLA method as given by equation (8.4.22), we have

$$\int_{\Omega^{(e)}} W_i^{(e)} \frac{\partial}{\partial H}\left[F_1(H)\left(\frac{\partial c^{(e)}}{\partial H}+\frac{\partial \bar{\phi}}{\partial H}c^{(e)}\right)\right]\partial\Omega$$
$$= \oint_{H=\delta} W_i^{(e)}\left(\frac{c_\delta^{(e)}}{\exp(-\bar{\phi}(\delta))\int_0^\delta \frac{\exp(\bar{\phi}(H))}{F_1(H)}dH}\right)\partial\Gamma$$
$$- \int_{\Omega^{(e)}} \frac{\partial W_i^{(e)}}{\partial H}\left[F_1(H)\left(\frac{\partial c^{(e)}}{\partial H}+\frac{\partial \bar{\phi}}{\partial H}c^{(e)}\right)\right]\partial\Omega \tag{8.5.25}$$

Substituting equation (8.5.25) into equation (8.5.23) yields

$$\int_{\Omega^{(e)}} W_i^{(e)}\left[\frac{\partial c^{(e)}}{\partial t}+\frac{\partial c^{(e)}}{\partial \bar{x}}\left(PeF_3(H)(H+1)(2-(H+1)A)\right)\right]\partial\Omega$$
$$- \oint_{H=\delta} W_i^{(e)}\left(\frac{c_\delta^{(e)}}{\exp(-\bar{\phi}(\delta))\int_0^\delta \frac{\exp(\bar{\phi}(H))}{F_1(H)}dH}\right)\partial\Gamma$$
$$+ \int_{\Omega^{(e)}} \frac{\partial W_i^{(e)}}{\partial H}\left[F_1(H)\left(\frac{\partial c^{(e)}}{\partial H}+\frac{\partial \bar{\phi}}{\partial H}c^{(e)}\right)\right]\partial\Omega = 0 \tag{8.5.26}$$

In this method, the primary variable $c^{(e)}$ in equation (8.5.26) is approximated as follows

$$c^{(e)} = \sum_{i=1}^{r} N_i^{(e)}(\bar{x}, H)c_i^{(e)}(t) \tag{8.5.27}$$

where $N_i^{(e)}(\bar{x}, H)$ is the element shape function and $c_i^{(e)}(t)$ is the nodal solution of the element at dimensionless time (t). Substituting equation (8.5.27) into equation (8.5.26) yields the following system of first order ordinary differential equations

$$\mathbf{M}^{(e)}\dot{\mathbf{c}}^{(e)} + (\mathbf{K_c}^{(e)} + \mathbf{K_d}^{(e)} + \mathbf{K}_\phi{}^{(e)} + \mathbf{K_K}^{(e)})\mathbf{c}^{(e)} = \mathbf{F}^{(e)} \tag{8.5.28}$$

where

$$
\begin{aligned}
M_{ij}^{(e)} &= \int_{\Omega^{(e)}} W_i^{(e)} N_j^{(e)} \ \partial\Omega, \\
K_{cij}^{(e)} &= \int_{\Omega^{(e)}} W_i^{(e)} \left(PeF_3(H)(H+1)(2-(H+1)A)\right) \frac{\partial N_j^{(e)}}{\partial \bar{x}} \partial\Omega, \\
K_{dij}^{(e)} &= \int_{\Omega^{(e)}} \frac{\partial W_i^{(e)}}{\partial H} F_1(H) \frac{\partial N_j^{(e)}}{\partial H} \partial\Omega, \\
K_{\phi ij}^{(e)} &= \int_{\Omega^{(e)}} \frac{\partial W_i^{(e)}}{\partial H} \left(F_1(H) \frac{\partial \bar{\phi}}{\partial H}\right) N_j^{(e)} \partial\Omega \\
K_{Kij}^{(e)} &= \left(- \oint_{H=\delta} W_{wi}^{(e)} K N_{wj}^{(e)} \partial\Gamma\right) \\
F_i^{(e)} &= 0,
\end{aligned}
\tag{8.5.29}
$$

are the mass matrix, stiffness matrix due to convection, stiffness matrix due to diffusion, stiffness matrix due to the total potential and the stiffness matrix due to the flux boundary condition on the wall, respectively. In the computational procedure, these matrices are formed using the Gauss integration technique. This is described in the next section. Meanwhile, $W_{wi}^{(e)}$ and $N_{wj}^{(e)}$ are the element weighting and shape functions appropriate for the wall surface (we have used the 1D linear element for these functions), and

$$
K = \frac{1}{\exp(-\bar{\phi}(\delta)) \int_0^{\delta} \frac{\exp(\bar{\phi}(H))}{F_1(H)} dH} \tag{8.5.30}
$$

representing the constant associated with the dimensionless flux on the wall according to the SFBLA method. Equation (8.5.31) is then assembled for all m elements to yield the following global system of first order (in time) ordinary differential equations

$$
\mathbf{M}\dot{\mathbf{c}} + (\mathbf{K_c} + \mathbf{K_d} + \mathbf{K}_\phi + \mathbf{K_K})\mathbf{c} = \mathbf{F} \tag{8.5.31}
$$

where

$$\mathbf{M} = \sum_{e=1}^{m} \mathbf{M}^{(e)}, \quad \mathbf{K_c} = \sum_{e=1}^{m} \mathbf{K_c}^{(e)}, \quad \mathbf{K_d} = \sum_{e=1}^{m} \mathbf{K_d}^{(e)}, \tag{8.5.32}$$
$$\mathbf{K}_\phi = \sum_{e=1}^{m} \mathbf{K}_\phi{}^{(e)}, \quad \mathbf{K_K} = \sum_{e=1}^{m} \mathbf{K_K}^{(e)}, \quad \mathbf{F} = \sum_{e=1}^{m} \mathbf{F}^{(e)}$$

8.5.1.2 Gauss Integration Technique to Form Matrices

This section describes the Gauss integration technique used to form the matrices given by equation (8.5.29). In this work, the Galerkin method is used in the FEM spatial discretization technique described in Section 8.5.1.1. In this method, the element weighting function $W_i^{(e)}$ in equation (8.5.29) takes the same form as the element shape function (i.e., $W_i^{(e)} = N_i^{(e)}$). Furthermore, using the Gauss integration technique, and in accordance with equation (8.5.29) (with $W_i^{(e)} = N_i^{(e)}$, i.e., the Galerkin FEM method), the matrices can be formed as follows

$$\begin{aligned}
\mathbf{M}^{(e)} &= \sum_{i=1}^{ngp}\sum_{j=1}^{ngp} w_i w_j \mathbf{N}^{(e)T}\mathbf{N}^{(e)}|\mathbf{J}^{(e)}| \\
\mathbf{K_c}^{(e)} &= \sum_{i=1}^{ngp}\sum_{j=1}^{ngp} w_i w_j \mathbf{N}^{(e)T}(\mathbf{N}^{(e)}\boldsymbol{\alpha})\frac{\partial \mathbf{N}^{(e)}}{\partial \bar{x}}|\mathbf{J}^{(e)}| \\
\mathbf{K_d}^{(e)} &= \sum_{i=1}^{ngp}\sum_{j=1}^{ngp} w_i w_j \frac{\partial \mathbf{N}^{(e)}{}^T}{\partial H}(\mathbf{N}^{(e)}\mathbf{F_1})\frac{\partial \mathbf{N}^{(e)}}{\partial H}|\mathbf{J}^{(e)}| \\
\mathbf{K}_\phi{}^{(e)} &= \sum_{i=1}^{ngp}\sum_{j=1}^{ngp} w_i w_j \frac{\partial \mathbf{N}^{(e)}{}^T}{\partial H}(\mathbf{N}^{(e)}\boldsymbol{\beta})\mathbf{N}^{(e)}|\mathbf{J}^{(e)}| \\
\mathbf{K_K}^{(e)} &= -\sum_{i=1}^{ngp} w_i \mathbf{N}_w^{(e)T} K \mathbf{N}_w^{(e)}|\mathbf{J}^{(e)}|
\end{aligned} \tag{8.5.33}$$

where ngp is the number of Gauss points used for the element, while w_i and w_j are the weights of the ij Gauss point. For each element, the column vector $\boldsymbol{\alpha}$ represents the value of the convection parameter $PeF_3(H)(H+1)(2-(H+1)A)$ at each node of the element (see equation (8.5.29)), while the column vector $\mathbf{F_1}$ represents the value

of F_1 at each node of the element. Meanwhile, the column vector $\boldsymbol{\beta}$ represents the value of $F_1(H)\frac{\partial\bar{\phi}}{\partial H}$ at each node of the element (see equation (8.5.29)). The vector $\mathbf{N}^{(e)}$ in equation (8.5.33) is the vector of the element shape functions, but is in terms of the Gaussian coordinate (γ, η). In this coordinate system, this vector for a 4-node isoparametric element is given by

$$\mathbf{N}^{(e)} = [N_1^{(e)} \;\; N_2^{(e)} \;\; N_3^{(e)} \;\; N_4^{(e)}] \tag{8.5.34}$$

where

$$\begin{aligned} N_1^{(e)} &= \frac{1}{4}(\gamma - 1)(\eta - 1) \\ N_2^{(e)} &= \frac{1}{4}(\gamma + 1)(1 - \eta) \\ N_3^{(e)} &= \frac{1}{4}(\gamma + 1)(\eta + 1) \\ N_4^{(e)} &= \frac{1}{4}(1 - \gamma)(\eta + 1) \end{aligned} \tag{8.5.35}$$

Meanwhile, $|J^{(e)}|$ is the determinant of a matrix defined by

$$\mathbf{J}^{(e)} = \begin{bmatrix} \frac{\partial\bar{x}}{\partial\gamma} & \frac{\partial H}{\partial\gamma} \\ \\ \frac{\partial\bar{x}}{\partial\eta} & \frac{\partial H}{\partial\eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial(N_i^{(e)}\bar{x}_i)}{\partial\gamma} & \frac{\partial(N_i^{(e)}H_i)}{\partial\gamma} \\ \\ \frac{\partial(N_i^{(e)}\bar{x}_i)}{\partial\eta} & \frac{\partial(N_i^{(e)}H_i)}{\partial\eta} \end{bmatrix} = \begin{bmatrix} \left[\frac{\partial N^{(e)}}{\partial\gamma}\right]\{\bar{x}\} & \left[\frac{\partial N^{(e)}}{\partial\gamma}\right]\{H\} \\ \\ \left[\frac{\partial N^{(e)}}{\partial\eta}\right]\{\bar{x}\} & \left[\frac{\partial N^{(e)}}{\partial\eta}\right]\{H\} \end{bmatrix} \tag{8.5.36}$$

where $\{\bar{x}\}$ and $\{H\}$ are the vectors of the nodal values of the dimensionless coordinates. Meanwhile, $\left[\frac{\partial N^{(e)}}{\partial\gamma}\right]$ and $\left[\frac{\partial N^{(e)}}{\partial\eta}\right]$ are the vectors of the nodal derivative of the element shape function given in equation (8.5.35) with respect to the Gaussian coordinate (γ, η) as follows

$$\left[\frac{\partial N^{(e)}}{\partial \gamma}\right] = \left[\frac{\partial N_1^{(e)}}{\partial \gamma} \quad \frac{\partial N_2^{(e)}}{\partial \gamma} \quad \frac{\partial N_3^{(e)}}{\partial \gamma} \quad \frac{\partial N_4^{(e)}}{\partial \gamma}\right] = \frac{1}{4}\left[(\eta-1) \quad (1-\eta) \quad (\eta+1) \quad -(\eta+1)\right] \tag{8.5.37}$$

$$\left[\frac{\partial N^{(e)}}{\partial \eta}\right] = \left[\frac{\partial N_1^{(e)}}{\partial \eta} \quad \frac{\partial N_2^{(e)}}{\partial \eta} \quad \frac{\partial N_3^{(e)}}{\partial \eta} \quad \frac{\partial N_4^{(e)}}{\partial \eta}\right] = \frac{1}{4}\left[(\gamma-1) \quad -(\gamma+1) \quad (\gamma+1) \quad (1-\gamma)\right]$$

The derivative of the element shape function ($\mathbf{N}^{(e)}$, given by equation (8.5.35)) with respect to the dimensionless coordinates $(\bar{x}, H)$ appearing in equation (8.5.33) can be obtained using Chain rule as follows

$$\begin{aligned} \frac{\partial N^{(e)}}{\partial \gamma} &= \frac{\partial N^{(e)}}{\partial \bar{x}} \times \frac{\partial \bar{x}}{\partial \gamma} + \frac{\partial N^{(e)}}{\partial H} \times \frac{\partial H}{\partial \gamma} \\ \frac{\partial N^{(e)}}{\partial \eta} &= \frac{\partial N^{(e)}}{\partial \bar{x}} \times \frac{\partial \bar{x}}{\partial \eta} + \frac{\partial N^{(e)}}{\partial H} \times \frac{\partial H}{\partial \eta} \end{aligned} \tag{8.5.38}$$

In matrix form, the above can be presented as follows

$$\begin{Bmatrix} \frac{\partial N^{(e)}}{\partial \gamma} \\ \\ \frac{\partial N^{(e)}}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial \bar{x}}{\partial \gamma} & \frac{\partial H}{\partial \gamma} \\ \\ \frac{\partial \bar{x}}{\partial \eta} & \frac{\partial H}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N^{(e)}}{\partial \bar{x}} \\ \\ \frac{\partial N^{(e)}}{\partial H} \end{Bmatrix} = \left[\ \mathbf{J}^{(e)}\ \right] \begin{Bmatrix} \frac{\partial N^{(e)}}{\partial \bar{x}} \\ \\ \frac{\partial N^{(e)}}{\partial H} \end{Bmatrix} \tag{8.5.39}$$

Therefore

$$\begin{Bmatrix} \frac{\partial N^{(e)}}{\partial \bar{x}} \\ \\ \frac{\partial N^{(e)}}{\partial H} \end{Bmatrix} = \left[\ \mathbf{J}^{(e)}\ \right]^{-1} \begin{Bmatrix} \frac{\partial N^{(e)}}{\partial \gamma} \\ \\ \frac{\partial N^{(e)}}{\partial \eta} \end{Bmatrix} \tag{8.5.40}$$

8.5.2 Time Integration by GS4-1 Framework

Once the governing equation of the particle deposition problem is written as a set of ordinary differential equations (i.e., equation (8.5.31)) after the spatial discretization described in Section 8.5.1.1, the next step in the computational procedure is to solve such set of equations using an appropriate time integration solver. Therefore, in this section, we apply the new GS4-1 time integrator framework to solve the problem and give solutions at each time level, and throughout the whole simulation period. This procedure can be done, in summary, as follows:

GS4-1 Framework for Time Integration of Particle Deposition Ordinary Differential Equation (ODE):

Consider semi-discretized linear transient convection-diffusion equation of the following form

$$\mathbf{M}\dot{\mathbf{c}} + \mathbf{K}\mathbf{c} = \mathbf{F} \tag{8.5.41}$$

where $\mathbf{M}$ is the mass matrix, and $\mathbf{K} = \mathbf{K_c} + \mathbf{K_d} + \mathbf{K}_\phi + \mathbf{K_K}$ is the stiffness matrix due to convection, diffusion, total potential, and the flux boundary condition on the wall, respectively, while $\mathbf{F}$ is the force vector (see equation (8.5.29)). Meanwhile, $\mathbf{c}$ and $\dot{\mathbf{c}}$ are the primary variable and its time derivative to be solved as follows

Given $\mathbf{c}_n$ and $\dot{\mathbf{c}}_n$ at time t_n, find $\mathbf{c}_{n+1}$ and $\dot{\mathbf{c}}_{n+1}$ at t_{n+1} from

$$\begin{aligned}
&\left\{\left(\frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)}\right)\frac{\mathbf{M}}{\Delta t}+\left(\frac{1}{1+\rho_\infty}\right)\mathbf{K}\right\}\mathbf{c}_{n+1}\\
=&\left\{\left(\frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)}\right)\frac{\mathbf{M}}{\Delta t}+\left(\frac{1}{1+\rho_\infty}\right)\mathbf{K}\right\}\mathbf{c}_n\\
&+\left\{\left(\frac{3+\rho_\infty+\rho_\infty^s-\rho_\infty\rho_\infty^s}{2(1+\rho_\infty)}\right)-1\right\}\mathbf{M}\dot{\mathbf{c}}_n-\mathbf{K}\mathbf{c}_n\\
&+\mathbf{F}_n+\left(\frac{1}{1+\rho_\infty}\right)(\mathbf{F}_{n+1}-\mathbf{F}_n)
\end{aligned} \tag{8.5.42}$$

with the expression for the associated update at the end of each time level

$$\dot{\mathbf{c}}_{n+1} = (1 + \rho_\infty^s) \left(\frac{\mathbf{c}_{n+1} - \mathbf{c}_n}{\Delta t} \right) - \rho_\infty^s \dot{\mathbf{c}}_n \tag{8.5.43}$$

8.6 Validation of the Model

Prior to performing the parametric studies, it is of significant importance to first validate the computational code to be used for this particular problem. Such validation process can be accomplished by comparing the numerical results to experimental data of the same problem available in the literature such as those given in [9, 122]. However, it is to be noted that such data was obtained for specific parameter values used in the deposition experiments. Often, some of these parameter values are not given in these papers and therefore, the comparison is subject to difference due to the unknown parameters.

For the purpose of validating the particle deposition model developed in this study, we solve the model described in Section 8.4 for the particular deposition problem considered in the work of Sjollema and Busscher [9], which considers the deposition of polystyrene latex particles toward polymethylmethacrylate (PMMA) in a parallel plate flow cell. Their work was chosen for the purpose of validating the particle deposition model presented in this study since the comprehensive problem parameters and experimental data are given in [9]. It is important to highlight here that, although the particle type considered in [9] is polystyrene latex, and not silica, the physics of the particle deposition problem remains the same. Therefore, the deposition problem and the numerical results discussed here are still representative of the silica particle deposition, at least qualitatively. Note that the comprehensive experimental parameters and data for silica is not yet available and is currently undertaken by another study which investigates the problem experimentally. For completeness of the present study, we present in Section 8.10 some results of silica particle deposition using limited experimental paramaters. However, for the purpose of validating the model and investigating the effects of the various physicochemical parameters on the rate of deposition, the particular deposition problem considered in the work of Sjollema and Busscher [9] will be used due to the comprehensive data set. We next describe this

problem and the values of the parameters used.

8.6.1 Model Parameters

In [9], two types of polystyrene latices with different surface charge densities were considered: (1) type UV-82, with diameter of 736 nm, and (2) UV-148, with a diameter of 820 nm, both of which having a material density (ρ_p) of $1.05 \times 10^3 kg/m^3$. These latex particles were suspended in potassium phosphate buffers at pH of 7.0 with ionic strengths ranging from 10 to 100 mM at ambient temperature, flowing with two different flow rates: $0.15ml/s$ ($Re = 4.0$) and $0.04ml/s$ ($Re = 1.1$), which represent laminar conditions. Although these Re numbers are not representative of the pipeline flow in a geothermal power plant (which is typically of turbulent nature), we recall that the objective here is to validate the particle deposition model presented in this work, which considers the deposition of particles in a laminar, isothermal, and incompressible flow. The idea of how the model can be developed further in the future to consider the case of turbulent flow is described briefly in Chapter 9.

As previously mentioned, for the purpose of validating the particle deposition model presented here, we have to rely on available experimental data such as those obtained by Sjollema and Busscher [9], which are described in this section. In their experiments, the values of the zeta potential for both particles as well as the PMMA substratum (wall) were determined as shown in Figure 8.4. The configuration of the parallel plate flow cell, as shown in Figure 8.5, was composed of two PMMA plates of size $5.5\,cm \times 3.8\,cm$ with separation distance of $0.06\,cm$. Additionally, a Hamaker constant (A_{132}) value of $7.0 \times 10^{21} J$ [9] was considered in the calculation of the Lifshitz-van der Waals interaction potential in the numerical model. Meanwhile, the dimensionless thickness of the diffusion boundary layer is given in [9] as $\delta = 0.018$. For validation purpose, we consider the results given in [9] corresponding to the deposition rate on the bottom plate of the cell at a particular location of $\bar{x} = 100$. It is however to be noted that the dynamic viscosity μ, density ρ_l and dielectric constant ϵ of the suspending medium (which is potassium phosphate buffers) were not given in [9], and therefore the values of these quantities are assumed to be those of water. The differences between the results given in [9] and our numerical results

are assumed to be due to these approximated values.

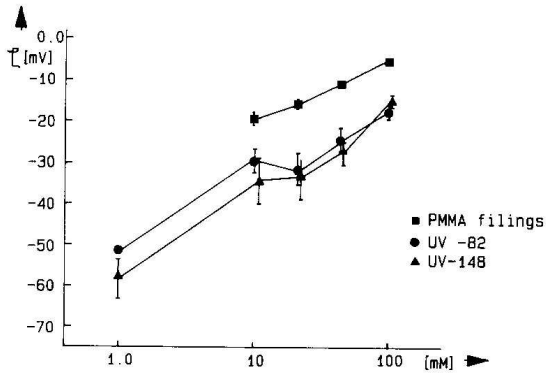


Figure 8.4: Zeta potentials of the polystyrene latex particles and the PMMA fillings suspended in potassium phosphate buffers at pH 7.0 as a function of the ionic strength value as given in [9] (bars denote the standard error of the mean value of the four independent measurements)

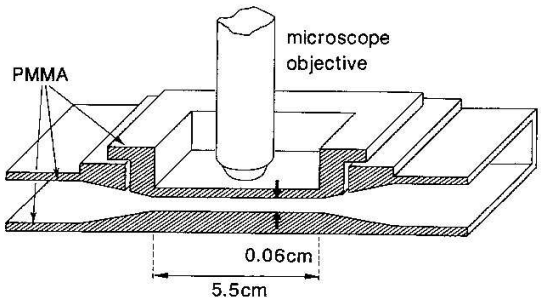


Figure 8.5: The configuration of the parallel plate flow cell employed in [9]

In our numerical model, the spatial domain is discretized (meshed) using commercial software ANSYS with 3661 4-node elements, giving a total of 3888 nodes for the UV-82 latex particle, and with 3859 4-node elements, giving a total of 4096 nodes for the UV-148 latex particle, both with refined mesh at regions close to the inlet and wall surface. The mass and stiffness matrices of these elements are formed using the Gauss integration technique described in Section 8.5.1.2. The model is then solved using the developed computational framework with a time step size of $\Delta t = 120s$ and end time of $t = 3600s$ at which the steady state has been attained.

8.6.2 Numerical Results

Table 8.6.1 shows the experimental and numerical results of the particle deposition problem considered in [9] in terms of dimensionless deposition rate Sh_0 at a specific location $\bar{x} = 100$, and comparison to the values obtained by our numerical results. As seen in this table, we were able to reproduce the results given in [9], which agree with the experimental values for about 50% of the cases considered. According to [9], such a discrepancy between the numerical and experimental results is most likely due to the inhomogeneities and roughness of the wall surface which can affect the Hamaker constant value. Furthermore, the large standard error in the zeta potential measured (see Figure 8.4) indicates another heterogeneity, which can also contribute to the increase in the deposition rate at low ionic strengths. For mesh sensitivity analysis, we solve the problem again using greater number of total elements for the spatial discretization (the meshes were again generated using commercial software ANSYS), i.e., with 6565 4-node elements, giving a total of 6802 nodes for the UV-82 latex particle, and with 8467 4-node elements, giving a total of 8716 nodes for the UV-148 latex particle. In this case, similar results as those shown in Table 8.6.1 were obtained hence the consistency. This provides some confidence in the numerical method used in this study to solve the particle deposition model. It is worthy to point out that although the model by Sjollema and Busscher [9] was not able to yield good approximation to the experimental results for low ionic strength values, the work is of great improvement over the previously developed SFBLA method of Bowen and co-workers [118, 122], which was not able to give good results for all cases considered under the unfavorable deposition condition (i.e., in the presence of repulsive electrical double layer interaction as in the present discussions).

8.7 Effect of Selective Control Feature

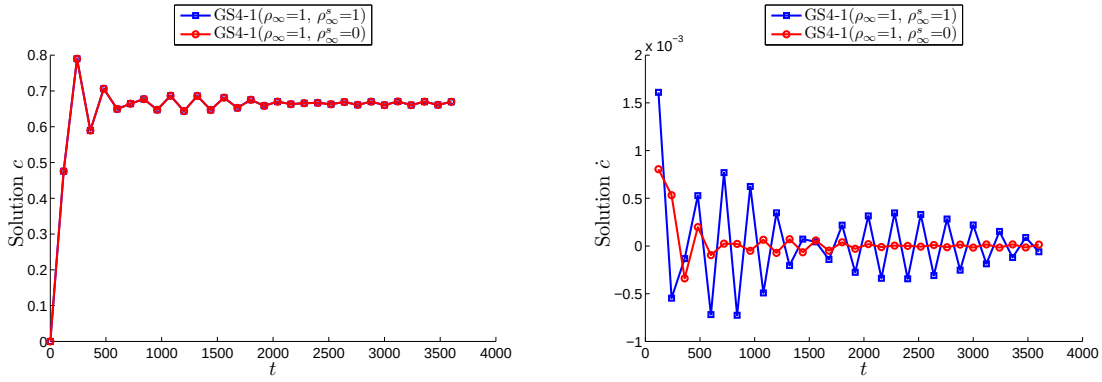
In this section, the effect of the selective control feature of the developed GS4-1 Framework is presented for the particle deposition problem considered in this chapter. Since the focus of this chapter is on the particle deposition problem itself, including its physics and the effects of the various flow and physicochemical parameters on the rate of deposition, less attention is given to the discussion on the effect of the selective control feature. The brief discussion of this feature is solely aimed to demonstrate the consistent algorithmic attributes of the GS4-1 framework.

Latex type	Flow rate (ml/s)	Ionic Strength (mM)	$Sh_{0,exp}$ $\times 10^{-3}$	$Sh_{0,Sjollema}$ $\times 10^{-3}$	$Sh_{0,present}$ $\times 10^{-3}$
UV-82	0.04	10	4.0	0	0
		20	7.0	0	0
		40	8.4	14.6	13.9
		100	7.8	14.6	13.9
	0.15	10	3.2	0	0
		20	10	0	0
		40	14	23	23
		100	7.6	23	23.4
UV-148	0.04	10	7.2	0	0
		20	8.4	0	0
		40	14	17	13.6
		100	14	17	14.5
	0.15	10	7.2	0	0
		20	15	0	0
		40	22	26	25
		100	9.9	26	24.7

Table 8.6.1: Comparisons of experimental and theoretical values of dimensionless initial deposition rate: $Sh_{0,exp}$ and $Sh_{0,Sjollema}$ represent the experimental and theoretical (numerical) Sherwood numbers (i.e., dimensionless initial deposition rate) as given in [9], while $Sh_{0,present}$ represents the numerical Sherwood number obtained in the current study

For this purpose, we solve the problem using the two cases, i.e., (i) GS4-1 framework with the selective control feature and (ii) GS4-1 framework without such a feature (see *Remark* 4.3.1). For each case, we solve the particle deposition model as described in Section 8.4 using the GS4-1 framework described and outlined in Section 8.4 with a time step size of $\Delta t = 120s$ and an end time of $t_{end} = 3600s$. Figure 8.6(a) and 8.6(b) show the dimensionless particle concentration and its time derivative generated by the two cases with $\rho_\infty = 1$, as a function of time (dimensionless) for a specific node at a location defined by coordinates $\bar{x} = 3.436, H = 0.018$. These figures show that both cases yield the same results on c, but the case without the selective control feature yields some numerical oscillations in the time derivative variable $\dot{c}$. The case with the selective control feature, on the other hand, yields relatively smaller numerical

oscillations. Attempt was then made to investigate the performance of the two cases when numerical damping is imposed on the primary variable. Figure 8.7 to 8.9 show the results generated by the two cases when ρ_∞ was chosen as 0.9, 0.8, 0.7, 0.6, 0.5, and 0.4. As seen in these figures, repeated and consistent observations on the effect of the selective control feature as those seen in the previous numerical examples were obtained for the particle deposition model considered in this chapter. This demonstrates the consistency of the arguments presented in the previous chapters.



(a) (b)

Figure 8.6: Plot of c and $\dot{c}$ as a function of time for node at $\bar{x} = 3.436, H = 0.018$ employing: (i) GS4-1 Framework with $\rho_\infty = \rho^s_\infty = 1$ (the case without selective control feature), and (ii) GS4-1 Framework with $\rho_\infty = 1, \rho^s_\infty = 0$ (the case with selective control feature)

8.8 Parametric Studies

In this section, we present the numerical results (in terms of dimensionless initial deposition rate Sh_0) employing the above particle deposition model to investigate the effects of the various flow and physicochemical parameters on the rate of the particle deposition. For this purpose, the variable parameters are: (1) the velocity of the flow (2) temperature of the carrying fluid, (3) the ionic strength of the solution, and (4) the particle size. Due to the lack of comprehensive data and parameter values of the deposition of silica particles, we used the values for the particle type employed in the work of Sjollema and Busscher [9] as this reference paper is the most comprehensive. For this parametric study, the particle deposition model described above was solved using the GS4-1 Framework with the selective control feature, in

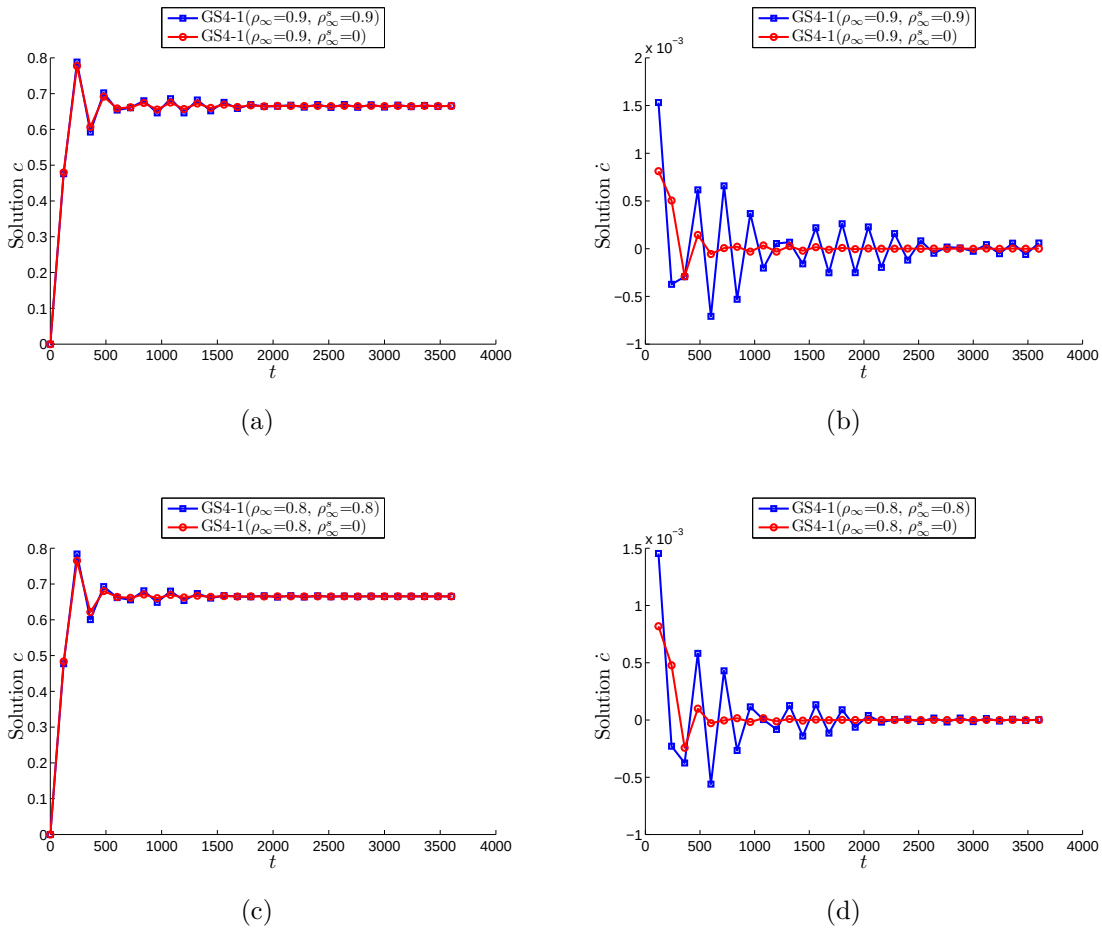


Figure 8.7: Plot of c and $\dot{c}$ as a function of time for node at $\bar{x} = 3.436, H = 0.018$ employing: (i) GS4-1 Framework with $\rho_\infty = \rho^s_\infty$ (the case without selective control feature), and (ii) GS4-1 Framework with $\rho^s_\infty = 0$ (the case with selective control feature), for ρ_∞ value of 0.9 and 0.8

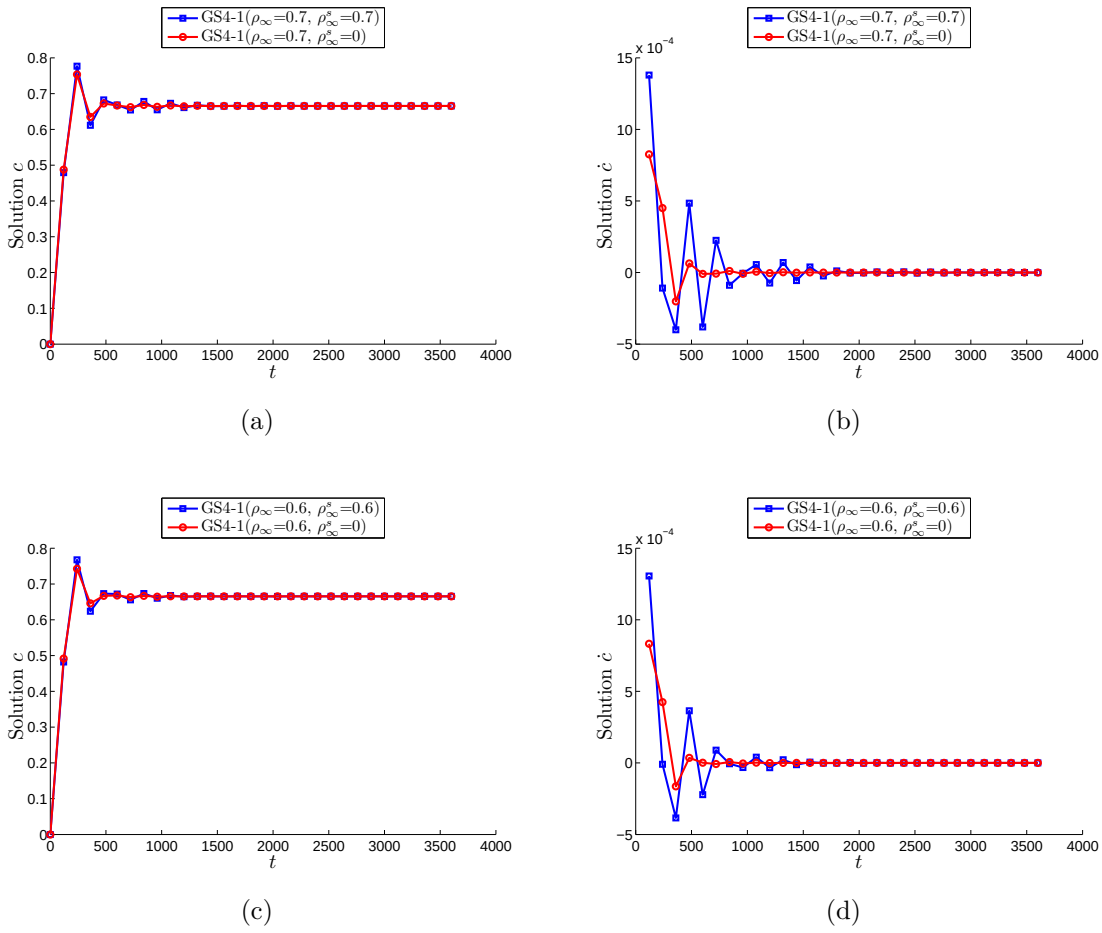


Figure 8.8: Plot of c and $\dot{c}$ as a function of time for node at $\bar{x} = 3.436, H = 0.018$ employing: (i) GS4-1 Framework with $\rho_\infty = \rho^s_\infty$ (the case without selective control feature), and (ii) GS4-1 Framework with $\rho^s_\infty = 0$ (the case with selective control feature), for ρ_∞ value of 0.7 and 0.6

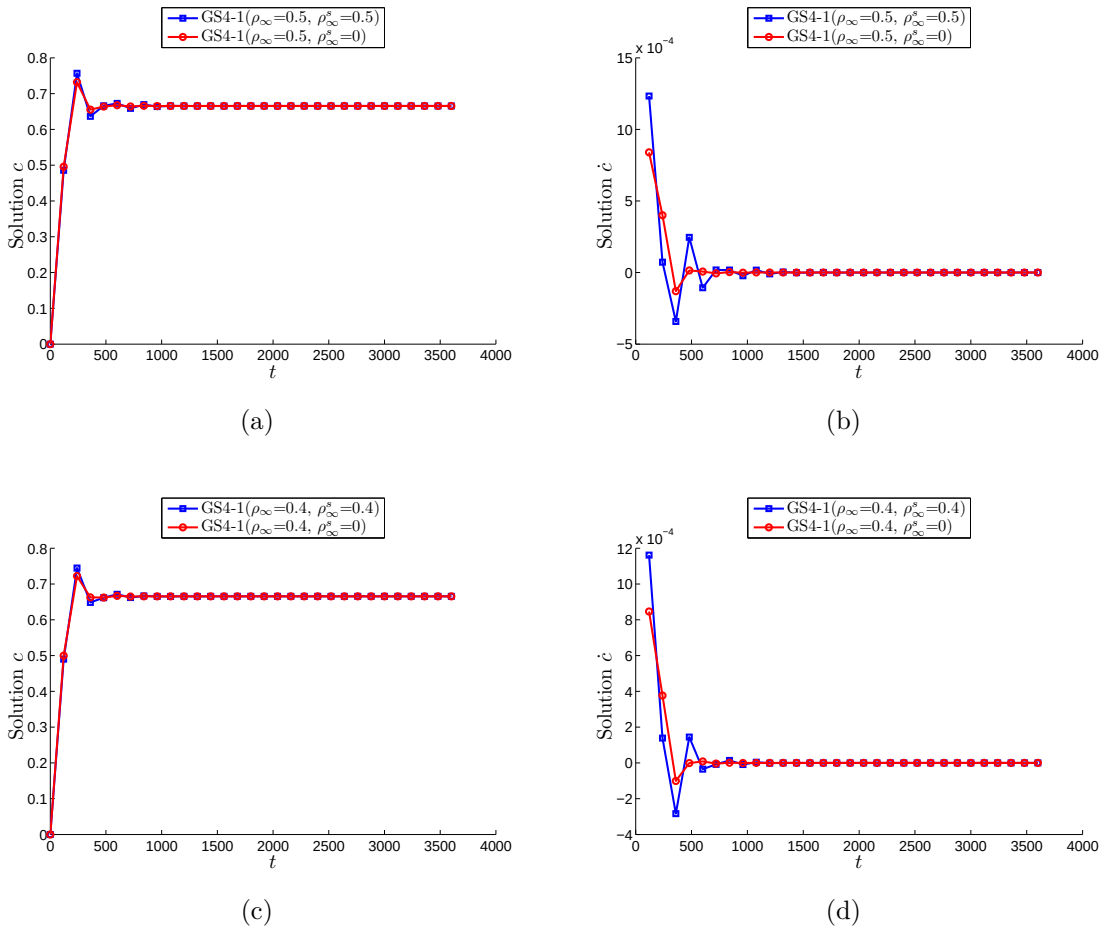


Figure 8.9: Plot of c and $\dot{c}$ as a function of time for node at $\bar{x} = 3.436, H = 0.018$ employing: (i) GS4-1 Framework with $\rho_\infty = \rho^s_\infty$ (the case without selective control feature), and (ii) GS4-1 Framework with $\rho^s_\infty = 0$ (the case with selective control feature), for ρ_∞ value of 0.5 and 0.4

particular with choice of $\rho_\infty = 0.9$ and $\rho^s_\infty = 0$, which was shown in Section 8.7 to yield satisfactory results.

8.8.1 Flow Velocity

We first look at the effect of the flow velocity. For this purpose, we consider, for the two types of particles considered in [9]: (a) UV-82, and (b) UV-148, for four colloidal suspensions of different flow velocities: $v = 0.004$, 0.04, 0.4 and $4ml/s$. For each case, the ionic strength of the solution is $C_s = 100mM$ while the temperature is chosen as $T = 25^oC$. For each case, we determine the dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ as given by equation (8.4.22) and show the result using a log-log plot as shown in Figure 8.10. Note that the straight lines in such log-log plots indicate a power law behaviour, i.e., $Sh_0 \propto \bar{x}^\alpha$, where the slope of the lines is the exponent of the power law α. As seen in Figure 8.10, α is obtained to be -0.3641 and -0.3737 for the particle types UV-82 and UV-148, respectively. This means that the deposition rate decreases as the colloidal suspension moves further away from the inlet (i.e., as $\bar{x}$ increases) and such a behaviour can be described by a power law with such exponent values. Additionally, this figure shows that, for both particle types, the deposition rate increases as the flow velocity increases. When the velocity at which the suspension flows is increased, the problem Peclet number (which represents the ratio of the rate of convection of the particles by the flow to the rate of diffusion of the particles driven by gradient of the particle concentration) is also increased as evident from equation (8.3.4). Table 8.8.2 shows the Peclet number for each case considered in Figure 8.10. How the Peclet number affects the rate at which the particles deposit on the surface can easily be seen from available/published analytical model of particle deposition that is typically employed in particle deposition studies to validate the numerical model. This analytical model, known as Smoluchowski-Levich approximation [99, 110], greatly simplifies the complex problem by assuming that the hydrodynamic interaction between the particles and the wall surface is counterbalanced by the van der Waals attraction between the two surfaces, and that all other colloidal interactions and external forces are absent/negligible. With such simplifying assumptions and under steady state condition, analytical solutions to the resulting governing equation can be derived and are avail-

able in the literature for some well-defined/ideal geometries, such as rotating disc or stagnation-point flow collector [99], plate in a uniform flow [99], cylinder in a uniform flow [99], parallel-plate or cylindical channels [100–102], and moving wire or plate [103]. In particular, for the initial stage of the particle deposition through parallel plate channel, the analytical solution (given in terms of the initial deposition rate) of this simplified model is given, at steady state, as [100, 106, 101, 102]

Latex type	Flow rate (ml/s)	Pe
UV-82	0.004	0.00022
	0.04	0.0022
	0.4	0.022
	4	0.22
UV-148	0.004	0.00034
	0.04	0.0034
	0.4	0.034
	4	0.34

Table 8.8.2: Peclet number for each case considered in Figure 8.10

$$Sh_0 = \frac{1}{\Gamma\left(\frac{4}{3}\right)}\left(\frac{2Pe}{9\bar{x}}\right)^{1/3} \tag{8.8.44}$$

where Sh_0 is the dimensionless initial deposition rate (i.e., the Sherwood number) and Pe is the problem Peclet number defined as (see also equation (8.3.4))

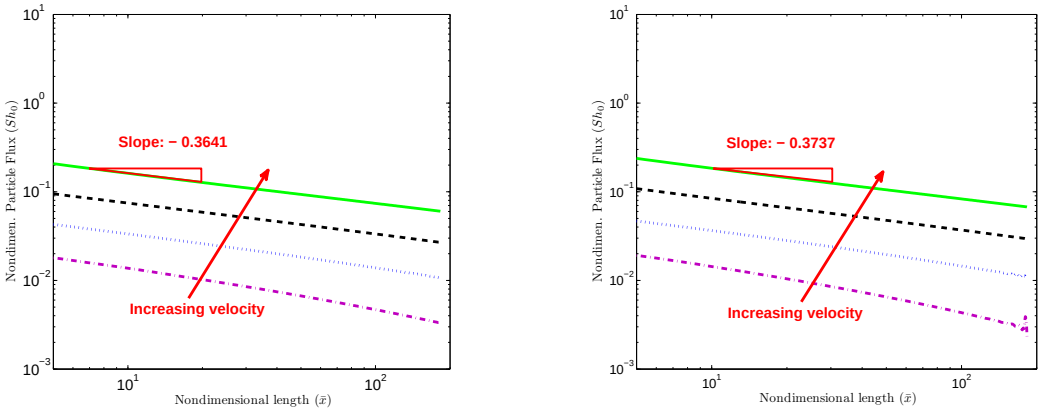
$$Pe = \frac{3V_m a^3}{2b^2 D_\infty} \tag{8.8.45}$$

Equation (8.8.44) indicates that the initial deposition rate increases as the Peclet number of the problem increases. This, therefore, justifies the observations seen in Figure 8.10. A close observation of this figure indicates for the same flow velocity the rate of deposition is relatively smaller for the particle type UV-82 (diameter, d, of $736\,nm$, see Figure 8.10(a)) than that for the particle type UV-148 ($d = 820\,nm$, see Figure 8.10(b)). This is due to the effect of the particle size. As shown in Table 8.8.2, the Peclet number for the particle type UV-148 ($d = 820\,nm$) is relatively larger than that for the particle type UV-82 ($d = 736\,nm$). Additionally, the particle

size also plays role in the physicochemical characteristics of the problem. Table 8.8.3 shows, for both particle types, the dimensionless Debye-Huckel reciprocal length τ (which plays significant role in the determination of the Electrical Double Layer potential ϕ_{EDL} as seen in equation (8.3.19)), the dimensionless Debye-Huckel length (which indicates the ϕ_{EDL} thickness, i.e., a length in the H direction at which ϕ_{EDL} plays role), and the resulting K value which dictates the rate of deposition according to Sjollema and Busscher [9] (see equation (8.4.22) and (8.5.30)). It is worthy to mention that for both particle types these physicochemical characteristics are the same regardless of the flow velocity values. From Table 8.8.3, it can be seen that the different particle size values yield different characteristics associated with the Electrical Double Layer potential ϕ_{EDL}, and consequently different values of the constant K. As the particle size increases, τ increases, and therefore the range at which ϕ_{EDL} plays role $(1/\tau)$ decreases. Since the repulsive potential acts over only a smaller length (in the H direction), the rate of deposition relatively increases as evident from the larger value of the K constant. It is also to be noted that the different particle sizes do not have any effect on the value of the Lifshitz-van der Waals interaction ϕ_{LW} since the evaluation of this attractive potential at a distance H is independent of the particle size (see equation (8.3.19)). Therefore, in terms of the physicochemical characteristics, the particle size has only effect on the ϕ_{EDL} repulsive potential. The effects of the particle size on both the problem Peclet number and the ϕ_{EDL} characteristics therefore explain the different results between the two particle types as observed in Figure 8.10. Additionally, we can see from this figure that the magnitude of the exponents (α, represented by the slope of the straight lines) of the power law behaviour, i.e., $Sh_0 \propto \bar{x}^{\alpha}$ for these suspensions are larger than that for the Smoluchowski-Levich approximation given by equation (8.8.44). Recall that this approximation derives such an analytical solution in the absence of any colloidal and hydrodynamic interactions. Therefore, the increase in the magnitude of the exponents for the cases shown in Figure 8.10 indicates the effect of such interactions in the power law behaviour describing the deposition rate. Furthermore, the two particle sizes also yield different exponent values as seen in Figure 8.10, which further indicates the effect of the particle size on these interactions and consequently on the rate of deposition.

Latex type	Diameter (nm)	Dimensionless Debye-Huckel reciprocal length (τ)	Dimensionless ϕ_{EDL} thickness ($1/\tau$)	K
UV-82	736	382.5901	0.0026	15.6186
UV-148	820	426.2553	0.0023	15.9095

Table 8.8.3: Physicochemical characteristics for the particle types considered in Figure 8.10



(a) (b)

Figure 8.10: Log-log plot of dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ for two cases: (a) UV-82, and (b) UV-148, for four colloidal suspensions of different flow velocities: $v = 0.004$, 0.04, 0.4 and $4ml/s$. Meanwhile, the ionic strength of the solution is $C_s = 100mM$ while the temperature is chosen as $T = 25^oC$.

Meanwhile, Figures 8.11(a) and 8.11(b) show the colloidal interaction potential ($\bar{\phi}$) as a function of dimensionless separation distance H for the two cases: (a) UV-82, and (b) UV-148 considered in Figure 8.10. These figures indicate one important information; that the colloidal interaction potentials involved in the particle deposition considered in this study is dominated by the Lifshitz-van der Waals interaction ϕ_{LW} potential. However, due to the different ranges of the ϕ_{EDL} and ϕ_{LW} with the latter being two orders of magnitude larger than the former, the different ϕ_{EDL} characteristics between the two particle types cannot be revealed from these figures. Therefore, in Figure 8.11(c) we plotted the ϕ_{EDL} potential alone, to illustrate the different ϕ_{EDL} characteristics between the two particle types due to the different sizes of the particles. An observation of Figure 8.11(c) indicates that the two particle types do yield

different ϕ_{EDL} characteristics as discussed previously.

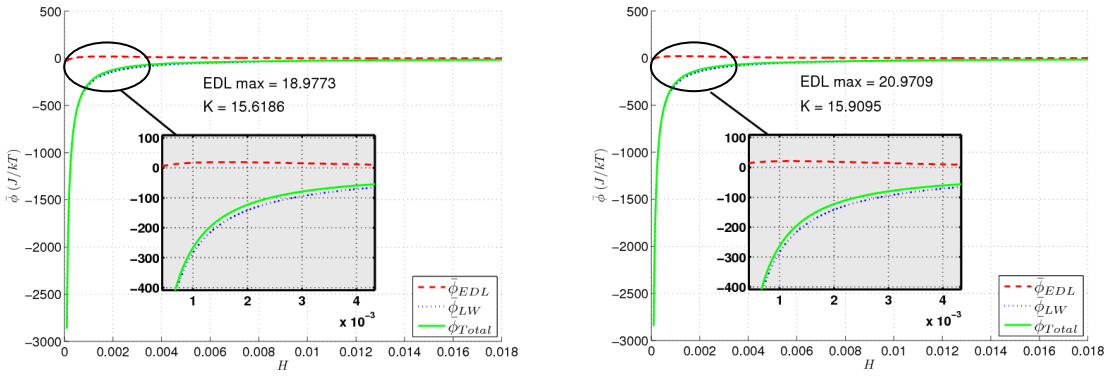
8.8.2 Temperature

Figure 8.12 shows the dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ for two particle types: (a) UV-82, and (b) UV-148, for four colloidal suspensions of different flow temperatures: $T = 25^oC, T = 50^oC, T = 70^oC$, and $T = 100^oC$. For each case, the ionic strength of the solution is chosen as $C_s = 100mM$ while the flow velocity is $v = 0.04ml/s$. This figure indicates, for both particle types, that the deposition rate reduces as the temperature increases. The temperature has effects on both the problem Peclet number and the physicochemical characteristics associated with the colloidal interaction potentials. Table 8.8.4 shows, for both particle types, the diffusion coefficient (D_∞) and the resulting Peclet number for each temperature. This table shows that as the temperature increases, the diffusion coefficient increases, and therefore the problem Peclet number reduces and subsequently so does the deposition rate. How the Peclet number influences the rate of deposition has been explained above.

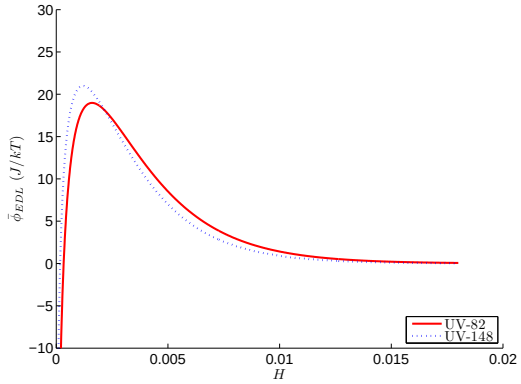
Latex type	$T(^oC)$	D_∞	Pe
UV-82	25	$6.6644\text{x}10^{-13}$	0.0022
	50	$1.1753\text{x}10^{-12}$	0.0012
	70	$1.6898\text{x}10^{-12}$	$8.6232\text{x}10^{-4}$
	100	$2.6327\text{x}10^{-12}$	$5.5351\text{x}10^{-4}$
UV-148	25	$5.9817\text{x}10^{-13}$	0.0034
	50	$1.0549\text{x}10^{-12}$	0.0019
	70	$1.5167\text{x}10^{-12}$	0.0013
	100	$2.3630\text{x}10^{-12}$	$8.5284\text{x}10^{-4}$

Table 8.8.4: Peclet number for each case considered in Figure 8.12

In addition, the temperature also affects the physicochemical characteristics of the problem as the dimensionless colloidal potentials are defined as $\bar{\phi} = \phi/kT$ (see equation (8.3.4)) where $\bar{\phi} = \bar{\phi}_{LW} + \bar{\phi}_{EDL}$. This means that the magnitude of $\bar{\phi}$ (used in the determination of K as shown in equation (8.5.30)) decreases as T increases. However, as discussed previously, the colloidal interaction potentials involved in the deposition problem considered in this study is dominated by the attractive

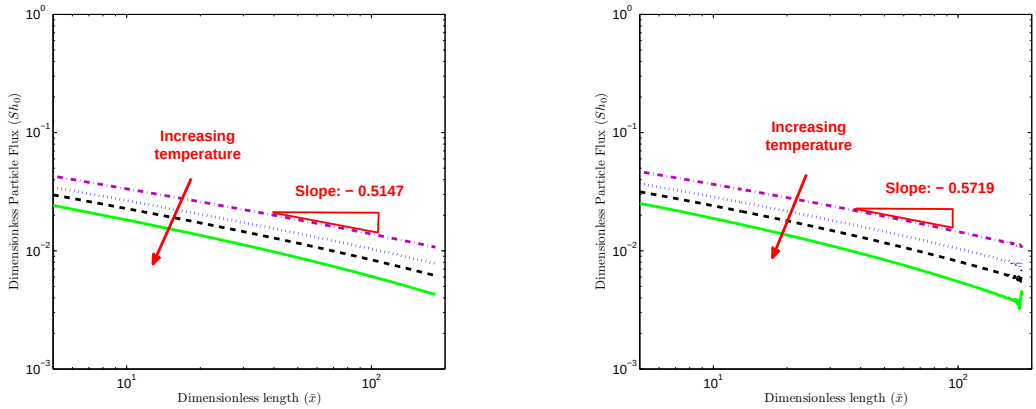


(a) (b)



(c)

Figure 8.11: Plot of colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for two cases: (a) UV-82 and (b) UV-148, considered in Figure 8.10, with ionic strength of $C_s = 100mM$ and temperature of $T = 25^oC$. Meanwhile, (c) shows the ϕ_{EDL} potential for the two cases



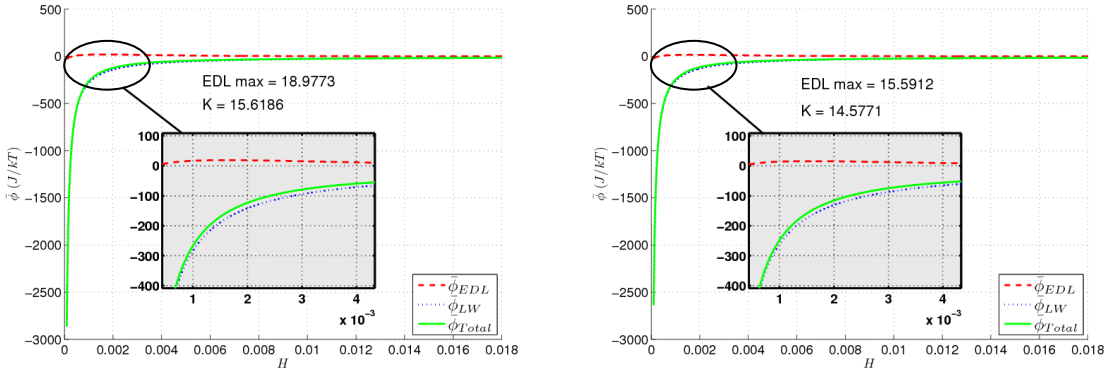
(a) (b)

Figure 8.12: Log-log plot of dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ for two cases: (a) UV-82, and (b) UV-148, for four colloidal suspensions of different flow temperatures: $T = 25^oC, T = 50^oC, T = 70^oC$, and $T = 100^oC$. Meanwhile, the ionic strength of the solution is chosen as $C_s = 100mM$ while the flow velocity is $v = 0.04ml/s$.

$\bar{\phi}_{LW}$ potential by two orders of magnitude. Therefore, as this potential decreases due to the increase in temperature, there is less attraction to the wall surface and consequently the rate at which deposition occurs reduces. This can also be explained from the mathematical point of view; hence the consistency. Since the dominant colloidal interaction potential, which is the Lifshitz-van der Waals interaction $\bar{\phi}_{LW}$, is attractive in nature and therefore is represented by a negative (-) sign (see equation (8.3.18)), the net potential $\bar{\phi} = \bar{\phi}_{LW} + \bar{\phi}_{EDL}$ thus is attractive and has a negative sign. This means that a smaller magnitude of this net potential (due to the increase in temperature) results in a larger value of the denominator of the constant K (see equation (8.5.30)), and consequently the value of K itself decreases. This indicates that the effect of temperature on the physicochemical characteristics of the deposition problem, as observed in Figure 8.12, can be explained from the physics point of view as well as from the mathematical/model perspective. Additionally, the exponents of the power law behaviour α for the suspensions considered in this figure are obtained to be -0.5147 and -0.5719 for the particle types UV-82 and UV-148, respectively.

Figure 8.13 and 8.14 shows the colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for particle types UV-82 and UV-148, respectively, at four temperatures considered in Figure 8.12. It is first important to note from these figures that for both particle types and for all temperature considered, the total/net colloidal interaction potential $\bar{\phi}_{total}$ is indeed dominated by the Lifshitz-van der Waals interaction $\bar{\phi}_{LW}$, which is attractive in nature and is represented by a negative (-) sign, as discussed above. Secondly, as also observed and discussed previously, the effect of temperature on the physicochemical characteristics of the problem cannot be revealed directly from such plots. It is only through the use of the flux constant K that such effect can be taken into account as evident from the different values of this constant for the different flow temperatures shown in Figure 8.13 and 8.14. To illustrate the effect of temperature on the net colloidal potential $\bar{\phi}$ which consequently influences the evaluation of the constant K as discussed above, we show plots of this net potential for all temperatures considered, as shown in Figure 8.15. An observation of Figure 8.15 indicates that as the temperature increases, the magnitude of the net potential $\bar{\phi}$ indeed decreases. As the net colloidal interaction potential is dominated by the attractive, negative $\bar{\phi}_{LW}$ potential by two orders of magnitude in comparison to the repulsive, positive $\bar{\phi}_{EDL}$, a smaller net potential $\bar{\phi}$ thus means less attraction to the wall surface and consequently reduces the rate of deposition. This justifies the observations seen in Figure 8.12.

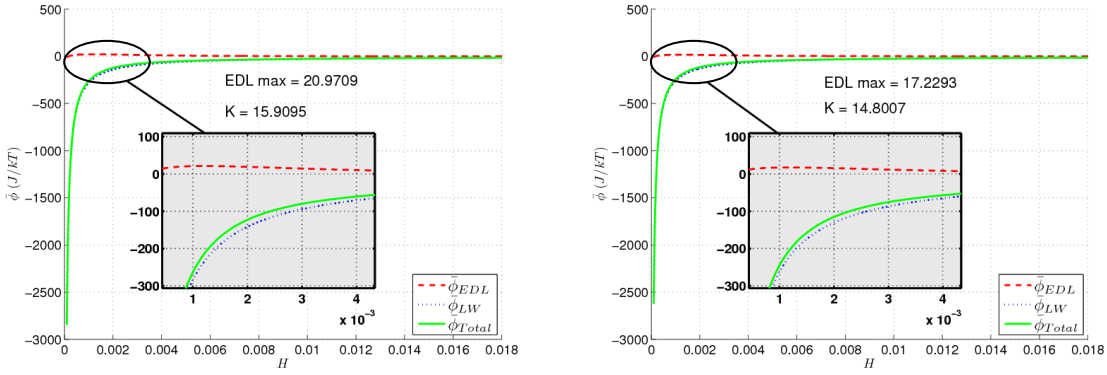
Additionally, a close observation of Figure 8.12 indicates that the two particle types have slightly different rate of initial deposition. For the same temperature value, the particle type UV-148 has larger value of Sh_0. This supports our previous discussion (in the parametric studies on flow velocity) that the particle size has effects on the rate of particle deposition. These effects are both in the flow conditions and physicochemical characteristics. To see these effects in the present case, we show in Table 8.8.5 the value of the diffusion coefficient, Peclet number, dimensionless Debye-Huckel reciprocal length τ (which plays significant role in the determination of the Electrical Double Layer potential ϕ_{EDL} as seen in equation (8.3.19)), the dimensionless Debye-Huckel length ($1/\tau$, which indicates the ϕ_{EDL} thickness, i.e., a length in the H direction at which ϕ_{EDL} plays a role), and the resulting K value which dictates the rate of deposition according to Sjollema and Busscher [9] (see



(a) $T = 25^oC$ (b) $T = 50^oC$

(c) $T = 70^oC$ (d) $T = 100^oC$

Figure 8.13: Plot of colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for particle type UV-82 at four colloidal suspensions of different flow temperatures: (a) $T = 25^oC$, (b) $T = 50^oC$, (c) $T = 70^oC$, and (d) $T = 100^oC$. Meanwhile, the ionic strength of the solution is $C_s = 100mM$ while the flow velocity is $v = 0.04ml/s$.



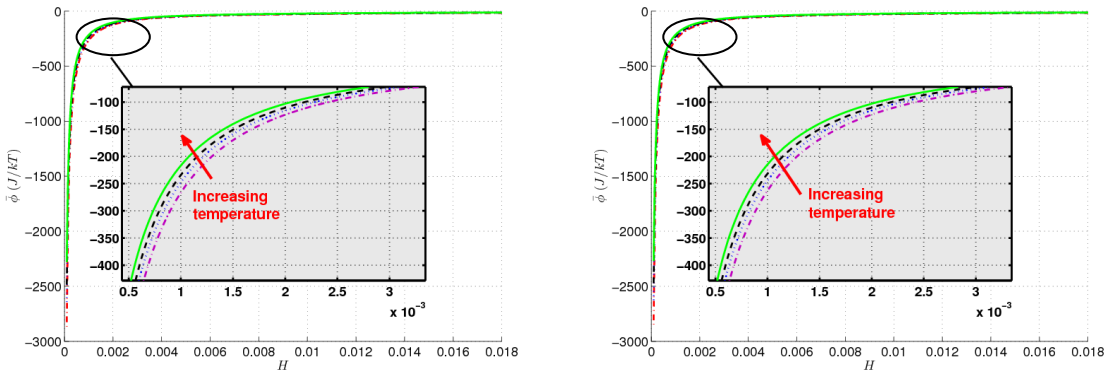
(a) $T = 25^oC$

(b) $T = 50^oC$

(c) $T = 70^oC$

(d) $T = 100^oC$

Figure 8.14: Plot of colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for particle type UV-148 at four colloidal suspensions of different flow temperatures: (a) $T = 25^oC$, (b) $T = 50^oC$, (c) $T = 70^oC$, and (d) $T = 100^oC$. Meanwhile, the ionic strength of the solution is $C_s = 100mM$ while the flow velocity is $v = 0.04ml/s$.



(a) UV-82 (b) UV-148

Figure 8.15: Plot of the net colloidal interaction potential $\bar{\phi} = \bar{\phi}_{LW} + \bar{\phi}_{EDL}$ as a function of dimensionless separation distance H for both particle types: (a) UV-82, and (b) UV-148 at four colloidal suspensions of different flow temperatures: $T = 25^oC$, 50^oC, 70^oC, and 100^oC. Meanwhile, the ionic strength of the solution is $C_s = 100mM$ while the flow velocity is $v = 0.04ml/s$.

equation (8.4.22) and (8.5.30)), for both particle types at the same temperature value. From this table, it can be seen that for the same temperature, the particle type UV-82 with relatively smaller radius has larger value of diffusion coefficient, and therefore smaller value of Peclet number as compared to those for the particle type UV-148. As the Peclet number Pe decreases, the rate of initial deposition Sh_0 also decreases (due to the relation between the Pe and Sh_0 discussed previously), hence explaining the observations seen in Figure 8.12. For the physicochemical characteristics, it can be seen from the table that for smaller particle size (UV-82), the dimensionless Debye-Huckel reciprocal length τ which plays significant role in the determination of the Electrical Double Layer potential ϕ_{EDL} as seen in equation (8.3.19) reduces, and therefore the dimensionless Debye-Huckel length ($1/\tau$, which indicates the ϕ_{EDL} thickness, i.e., a length in the H direction at which ϕ_{EDL} plays role) increases. Consequently, the constant K decreases contributing to the decline in the deposition rate as compared to those for the larger particle size (UV-148). It is to be noted from this table that the trends of differences in these flow and physicochemical characteristics between the two particle types are consistent for all values of temperature considered in this study. It is also to be recalled, as mentioned previously, that the different particle sizes do not have any effect on the value of

the Lifshitz-van der Waals interaction ϕ_{LW} since the evaluation of this attractive potential at a distance H is independent of the particle size (see equation (8.3.19)). The effects of the particle size on both the problem Peclet number and the ϕ_{EDL} characteristics therefore explain the different results between the two particle types as observed in Figure 8.12, which is also consistent with those observed in Figure 8.10.

Temp (^oC)	Characteristics	UV-82 ($d = 736\,nm$)	UV-148 ($d = 820\,nm$)
25	D_∞	6.6644x10^{-13}	5.9817x10^{-13}
	Pe	0.0022	0.0034
	τ	765.1456	852.4720
	$1/\tau$	0.0013069	0.0011731
	K	15.6186	15.9095
50	D_∞	1.1753x10^{-12}	1.0549x10^{-12}
	Pe	0.0012	0.0019
	τ	778.8352	867.7240
	$1/\tau$	0.0012840	0.0011524
	K	14.5771	14.8007
70	D_∞	1.6898x10^{-12}	1.5167x10^{-12}
	Pe	8.6232x10^{-4}	0.0013
	τ	792.0832	882.4840
	$1/\tau$	0.0012625	0.0011332
	K	13.8473	14.0266
100	D_∞	2.6327x10^{-12}	2.3630x10^{-12}
	Pe	5.5351x10^{-4}	8.5284x10^{-4}
	τ	814.8256	907.8220
	$1/\tau$	0.0012272	0.0011015
	K	12.8860	13.0133

Table 8.8.5: Flow and physicochemical characteristics for both particle types at various temperature considered in Figure 8.12

8.8.3 Ionic Strength

Finally, we discuss the effect of the ionic strength of the solution on the rate of deposition. For this purpose, we consider four colloidal suspensions of different ionic strength: $C_s = 10$, 20, 70, and 100 mM. For each case, the flow velocity is chosen as $v = 0.04ml/s$ while the temperature is $T = 25^oC$. The numerical results obtained show that, for both particle sizes, the resulting Sh_0 for $C_s = 70mM$ is the same

as those for $C_s = 100mM$, while for $C_s = 10mM$ and $20mM$, $Sh_0 = 0$ (for both particle sizes). Therefore, we only show the results for $C_s = 100mM$ as seen in Figure 8.16. From these results, it can be seen that in general the rate of initial deposition increases as the value of the ionic strength increases. The ionic strength influences the value of the dimensionless Debye-Huckel reciprocal length τ, which plays significant role in the determination of the Electrical Double Layer potential ϕ_{EDL} as seen in equation (8.3.19) and consequently affecting the value of the constant K and the resulting deposition rate Sh_0. As the ionic strength increases, τ increases, and therefore the dimensionless Debye-Huckel length $(1/\tau)$ decreases. Consequently, the constant K increases, which then increases the rate at which the particles deposit on the wall surface. It is also important to note that the ionic strength does not only influence the ϕ_{EDL} thickness, but also the magnitude of this repulsive force. As the ionic strength increases, the magnitude of the zeta potential ζ decreases as seen in Figure 8.4, and therefore the magnitude of the repulsive ϕ_{EDL} potential also decreases (see equation (8.3.19)). As both the magnitude and range of the repulsive ϕ_{EDL} potential decrease, the energy barrier, which hinders deposition due to the presence of a repulsive force, decreases and therefore the particles deposit on the wall surface at a higher rate. On the other hand, when the ionic strength is low, both the magnitude and range of the repulsive ϕ_{EDL} potential increase, and therefore the energy barrier is higher resulting in smaller rate of deposition. The effect of the ionic strength on ϕ_{EDL}, as discussed here, can be seen in Figure 8.17 and 8.18 for both particle types, UV-82 and UV-148, respectively, where we show the plot of colloidal potential $(\bar{\phi})$ as a function of the dimensionless separation distance H at the four suspension ionic strengths. Additionally, it is to be noted that the ionic strength does not influence the value of the Lifshitz-van der Waals interaction ϕ_{LW} (see equation (8.3.18)) as evident from Figures 8.17 and 8.18. It is also important to note that the ionic strength does not have any effect on the flow characteristics as the problem Peclet number is independent of such parameter (see equation (8.3.4)).

A close observation of Figure 8.16 indicate that the different particle sizes result in different rate of deposition Sh_0. Additionally, they also yield different exponents of the power law behaviour α (i.e., -0.4270 and -0.4603 for the particle types UV-82 and UV-148, respectively). This is consistent with our previous discussion that the

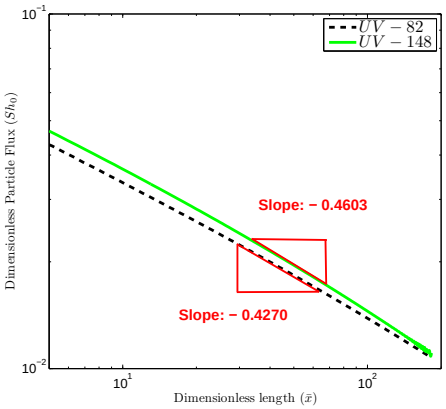
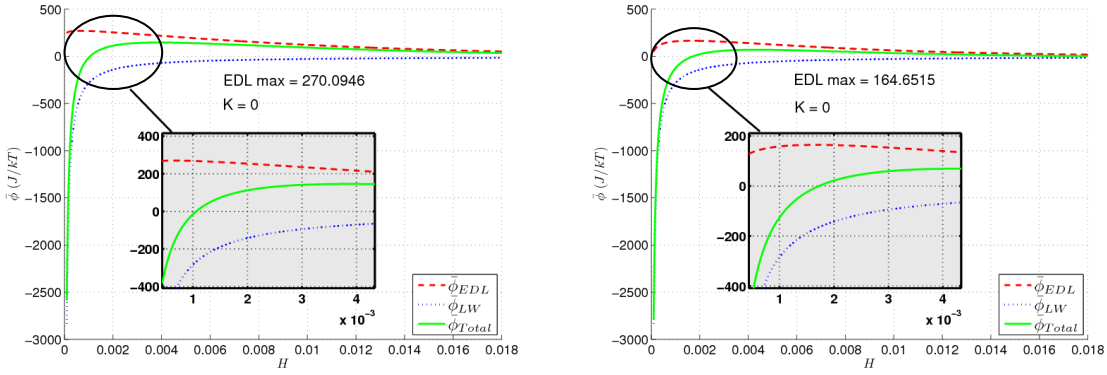


Figure 8.16: Log-log plot of dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ for both particle sizes (UV-82 and UV-148) for ionic strength $C_s = 100mM$. Note that, for both particle sizes, the resulting Sh_0 for $C_s = 70mM$ is the same as those for $C_s = 100mM$, while for $C_s = 10mM$ and $20mM$, $Sh_0 = 0$ (for both particle sizes). For all cases, the flow velocity is $v = 0.04ml/s$ while the temperature is $T = 25^oC$.

particle size influences the deposition rate. To see this effect more clearly, we show in Table 8.8.6 the flow and physicochemical characteristics of both particle types for all values of ionic strength considered in Figure 8.16. From this table, we can see that the particle size effects both the flow and physicochemical characteristics of the problem, the latter being associated with the repuslive $\bar{\phi}_{EDL}$ potential alone, as the attractive $\bar{\phi}_{LW}$ potential is independent of the ionic strength value. In terms of the flow condition, it can be seen from Table 8.8.6 that as the particle size increases, the diffusion coefficient decreases, and therefore the Peclet number increases. This results in higher rate of deposition due to the functional relation between the two as discussed previously. Additionally, in terms of the physicochemical characteristics, the dimensionless Debye-Huckel reciprocal length τ increases as the particle size increases, and therefore the dimensionless Debye-Huckel length $(1/\tau)$ decreases. Consequently, the constant K increases, which then increases the rate at which the particles deposit on the wall surface. This trend can be seen in the cases with relatively higher ionic strength $(C_s = 70, 100mM)$ as shown in Table 8.8.6. However, the other cases with low values of ionic strength $(C_s = 10, 20mM)$, both result in value of $K = 0$. For these two cases, the resulting energy barrier is relatively higher, with a wider range as seen in Figure 8.17 and 8.18, for both particle types UV-82 and UV-148, respectively.



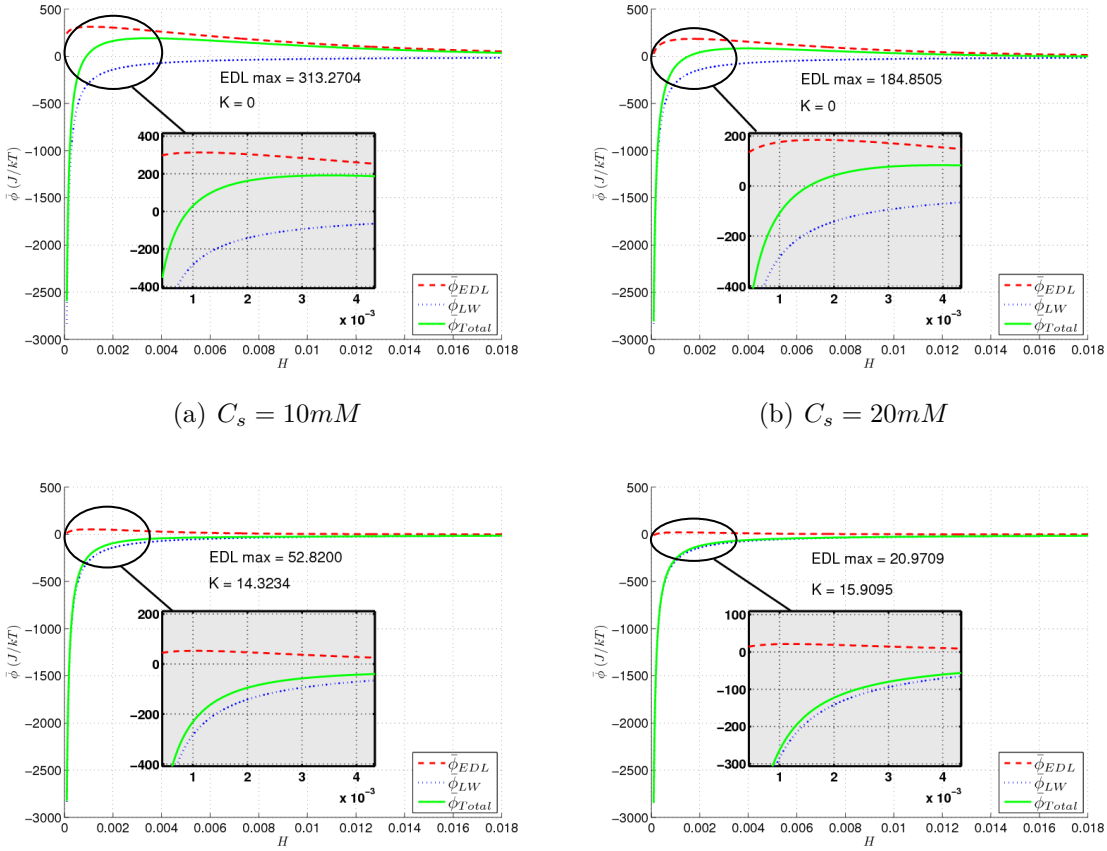
(a) $C_s = 10mM$

(b) $C_s = 20mM$

(c) $C_s = 70mM$

(d) $C_s = 100mM$

Figure 8.17: Plot of colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for particle type UV-82 at four colloidal suspensions of different ionic strength: (a) $C_s = 10mM$, (b) $C_s = 20mM$, (c) $C_s = 70mM$, and (d) $C_s = 100mM$. Meanwhile, the flow velocity is $v = 0.04ml/s$ while the temperature is $T = 25^oC$.



(a) $C_s = 10mM$

(b) $C_s = 20mM$

(c) $C_s = 70mM$

(d) $C_s = 100mM$

Figure 8.18: Plot of colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for particle type UV-148 at four colloidal suspensions of different ionic strength: (a) $C_s = 10mM$, (b) $C_s = 20mM$, (c) $C_s = 70mM$, and (d) $C_s = 100mM$. Meanwhile, the flow velocity is $v = 0.04ml/s$ while the temperature is $T = 25^oC$.

Ionic Strength (mM)	Characteristics	UV-82 ($d = 736\,nm$)	UV-148 ($d = 820\,nm$)
10	D_∞	6.6644x10^{-13}	5.9817x10^{-13}
	Pe	0.0022	0.0034
	τ	120.7859	134.5712
	$1/\tau$	0.0082791	0.0074310
	K	0	0
20	D_∞	6.6644x10^{-13}	5.9817x10^{-13}
	Pe	0.0022	0.0034
	τ	341.6365	380.6276
	$1/\tau$	0.0029271	0.0026272
	K	0	0
70	D_∞	6.6644x10^{-13}	5.9817x10^{-13}
	Pe	0.0022	0.0034
	τ	640.1949	713.2606
	$1/\tau$	0.0015620	0.0014020
	K	13.3631	14.3234
100	D_∞	6.6644x10^{-13}	5.9817x10^{-13}
	Pe	0.0022	0.0034
	τ	765.1456	852.4720
	$1/\tau$	0.0013069	0.0011731
	K	15.6186	15.9095

Table 8.8.6: Flow and physicochemical characteristics for both particle types at various ionic strength considered in Figure 8.16

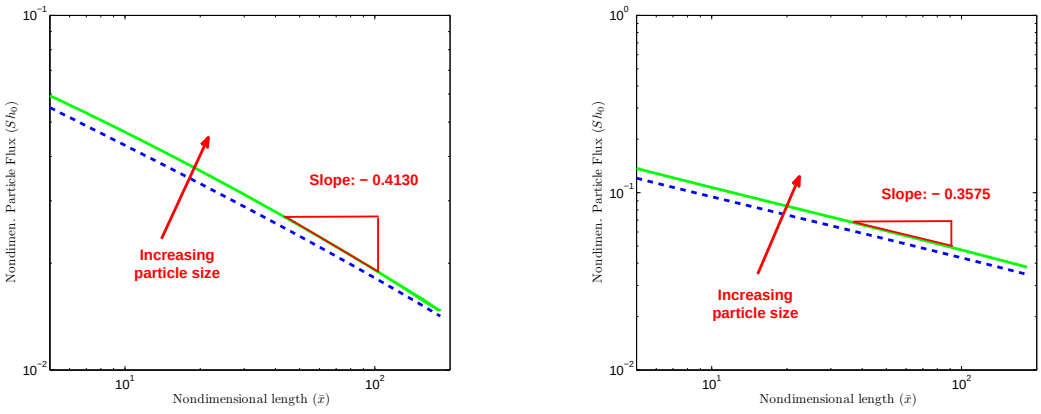
In these cases, the net colloidal interaction potential $\bar{\phi}$ is mostly repulsive in nature; i.e., the repulsive $\bar{\phi}_{EDL}$ is more dominant that the attractive $\bar{\phi}_{LW}$ potential (except for regions very close to the wall surface $H < 0.001$). The stronger repulsive force hinders the deposition to occur, as represented by the value of $K = 0$ in both cases. Moreover, it is to be noted from the table that, for the same type of particle, the diffusion coefficient D_∞ and Peclet number Pe are the same regardless of the value of the ionic strength. This proves that the ionic strength of the solution does not influence the flow characteristics of the problem as we discussed above.

8.8.4 Particle Size

The effect of particle size had been discussed above in Sections 8.8.1, 8.8.2 and 8.8.3 concurrently with the other parameters, i.e., flow velocity, temperature, and solution

ionic strength, respectively. Therefore, in this section, only conclusions about these effects will be presented.

As shown and discussed above, the particle size has effect in both the flow conditions and physicochemical characteristics. In a nutshell, particles with larger size will have higher rate of deposition, for the same value of flow velocity, temperature and solution ionic strength. For completeness, we show in Figure 8.19 comparison of the rate of deposition between the two particle types used in this study, i.e., UV-82 and UV-148, for two additional deposition cases not yet considered above.



(a) (b)

Figure 8.19: Log-log plot of dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ for both particle types: UV-82, and UV-148, for two additional colloidal suspension: (a) $v = 0.15ml/s$, $C_s = 70mM$, $T = 50^oC$, and (b) $v = 1.5ml/s$, $C_s = 70mM$, $T = 50^oC$

8.9 Recommendations Based on Results of Parametric Studies

The main purpose of undertaking the parametric studies discussed in Section 8.8 is to gain more understanding of the effects of the various flow and physicochemical parameters on the rate of the deposition. With such an understanding, more insights into mitigating the problem can be obtained, and from here, more effective approaches to tackle the problem can be taken. We present in this section a few recommendations for mitigating the deposition of particles for the particle deposition case considered in this study (i.e., laminar, isothermal flow, with monodispersed

colloidal particles). However, it is important to note that these recommendations are solely based on the model developed and the parametric studies done in the present study.

Based on the results of the parametric studies as shown and discussed in Section 8.8, we conclude that the rate of initial deposition of particles can be reduced if

- the particle size is smaller
- the flow velocity is reduced
- the flow temperature is increased
- the ionic strength is reduced

It is, however, important to mention that as the temperature of the flow/solution is increased further, the suspension may be subject to additional contribution to the deposition of the particles due to a phenomenon, called thermophoresis, due to the presence of temperature gradient in such case. This process is not considered in the present particle deposition model as we have only considered particle deposition from suspension with isothermal flow in this preliminary work. Therefore, the effect of temperature gradient has not been taken into account in this study. As the temperature of the brine at the geothermal power plant can go higher, it is recommended that any future work in this study takes such process into account for obtaining more reliable and representative results.

8.10 Numerical Results of Silica Particle Deposition

As previously mentioned, the above analysis on validating the particle deposition model as well as the parametric studies of the effect of the various flow and physicochemical parameters on the rate of deposition have focused on the deposition of polystyrene latex particles toward polymethylmethacrylate (PMMA) in a parallel plate flow cell as described in Sjollema and Busscher [9]. This is due to the availability of the comprehensive problem parameters and experimental data for such a particle

type. On the other hand, the comprehensive experimental parameters and data for the actual particle type of interest, which is silica, are not yet available and is currently undertaken by another study which investigates the problem experimentally.

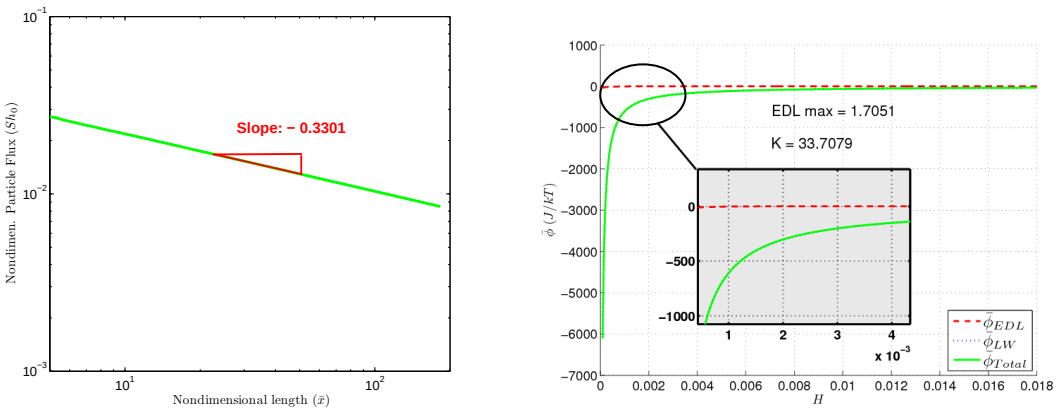
For completeness of the present study, we present in this section some preliminary results of silica particle deposition using limited experimental paramaters[1]. These parameters are: particle size $a = 76.3nm$, material density $\rho_p = 1.5 \times 10^3 kg/m^3$, electrical potential (i.e., the zeta potential at shear plane) of the particle $\zeta_1 = -3.25mV$ and the wall (mild steel) $\zeta_2 = -22.9mV$, respectively, electrolyte molar concentration (ionic strength of the solution) $C_s = 3.57\,M$, Hamaker constant typical for silica $A_{132} = 1.5 \times 10^{20} J$ [129], flow temperature $T = 25^oC$, and flow velocity $v = 4\,ml/s$. Meanwhile, the suspending medium is chosen as water with dynamic viscosity $\mu = 0.89 \times 10^{-3}\,Ns/m2$, density $\rho_l = 1000\,kg/m^3$ and dielectric constant $\epsilon = 78.8$. The problem is solved using the computational procedures described in Section 8.5. The spatial domain is discretized (meshed) using ANSYS with 6565 4-node elements, giving a total of 6802 nodes, with refined mesh at regions close to the inlet and wall surface. Meanwhile, a time step size of $\Delta t = 120s$ and end time of $t = 3600s$ are used in the time integration procedure at which the steady state has been attained.

Figure 8.20(a) shows the plot of the dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ while Figure 8.20(b) shows the corresponding colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for the silica particle deposition given the above values of the flow and physicochemical parameters. A comparison of Figures 8.10 and 8.20(a) indicates that the initial deposition rate (Sh_0) and magnitude of the exponent of the power law behaviour (α) for the silica particles considered here are lower than those for the polystyrene latex particles, although both cases have the same flow velocity of $v = 4\,ml/s$. This is largely due to the smaller size of the silica particles ($a = 76.3\,nm$) as compared to those of the polystyrene latex particles (diameter of $736\,nm$ for type UV-82 and diameter of $820\,nm$ for type UV-148). In terms of the flow characteristics, this smaller size of the silica particles results in a much lower problem Peclet number

[1]as given by Pavlo Kokhanenko, another Ph.D student who is currently investigating the problem experimentally

($Pe = 4.04 \times 10^{-4}$) as compared to the polystyrene latex particles ($Pe = 0.22$ for type UV-82 and $Pe = 0.34$ for type UV-148). With smaller Peclet number, the initial deposition rate (Sh_0) also reduces as discussed above.

On the other hand, in terms of the physicochemical characteristics of the problem, the much stronger ionic strength of the solution in which the silica particles are suspended results in a larger value of the constant K (shown in Figure 8.20(b)) in comparison to the values for the deposition of polystyrene latex particles (Section 8.8). As previously discussed in Section 8.8.3, the ionic strength influences the value of the dimensionless Debye-Huckel reciprocal length τ and consequently affects the value of the constant K as well as the resulting deposition rate Sh_0. As the ionic strength increases, τ increases, and therefore the dimensionless Debye-Huckel length ($1/\tau$) decreases. Consequently, the constant K increases, which then supposedly increases the rate at which the particles deposit on the wall surface. It is also important to recall that the ionic strength also influences the magnitude of the repulsive ϕ_{EDL} potential. As the ionic strength increases, the magnitude of the repulsive ϕ_{EDL} potential also decreases (see equation (8.3.19) and also Figure 8.20(b)). As both the magnitude and range of the repulsive ϕ_{EDL} potential decrease, the energy barrier decreases and therefore the particles deposit on the wall surface at a higher rate. However, as highlighted above, the silica particle deposition considered in this section has a much smaller Peclet number due to the particle size being much smaller; hence reducing the initial deposition rate (Sh_0). As a consequence, the initial deposition rate does not increase much despite the increase in the ionic strength. The results obtained here indicate that the particle size can have a more prominent influence on the rate of deposition as compared to the ionic strength.



(a) (b)

Figure 8.20: Plot of (a) dimensionless initial deposition rate (Sh_0) as a function of dimensionless length $\bar{x}$ (log-log plot), and (b) colloidal potential ($\bar{\phi}$) as a function of dimensionless separation distance H for the silica particle deposition

Chapter 9

Concluding Remarks and Future Directions

This final chapter highlights the fundamental contributions of the present research and suggests ideas and directions for future research in this line of approach.

9.1 Concluding Remarks

The primary goal of the research presented in this thesis was to develop a new time integration framework, under the class of single step single solve algorithms, for use in first order dynamic systems typical of those encountered in transient heat conduction, flow transport, and fluid dynamic problems, with computationally attractive features which overcome the restrictions in the existing and current state-of-the-art methods, and therefore contributes towards improving the state-of-the-art of such class of algorithms. In this thesis, we presented the theoretical developments via utilizing in a consistent manner the Algorithms by Design procedure, leading to the design of the new time integration framework, termed GS4-1, with computationally attractive features including second-order accuracy, unconditional stability, zero-order overshoot behaviour, and controllable numerical dissipation with the new selective control feature.

Throughout the thesis, we demonstrated, using various types of numerical examples

(linear and nonlinear) pertaining to first order transient systems, the ability and advantage of the newly developed GS4-1 framework in comparison to the existing methods. In particular, we demonstrate the roles played by the new selective control feature in generating physically accurate solutions of both the primary variable and its time derivative that is important to correctly capture the physics and dynamics of the problems, in contrast to the existing method without such a feature. The results from all the numerical examples considered in this study consistently indicated that the time derivative variable often requires more numerical damping to stabilize the numerical solution than the primary variable does. Without the selective control feature, the numerical solutions of this variable is often oscillatory and therefore does not represent the dynamics of the problem correctly. On the other hand, this requirement can easily be met via the selective control feature available in the present two-parameter GS4-1 framework, which provides a more flexible and selective control of the numerical damping associated with the two variables.

Not only that this new feature improves the numerical solution of the time derivative variable, but also it plays a role in improving the numerical solution of the primary variable, in comparison to the existing methods without such a feature. Additionally, it provides the necessary avenue to achieve this without imposing over-dissipation, which may yield numerical solutions that, although are stable, deviate from the actual dynamics of the problem as demonstrated in this thesis. Therefore, the significances of this new feature and the abilities of the newly developed framework in comparison to existing methods are successfully demonstrated. In this regard, this new feature serves as an added value and is a key desirable feature of the overall GS4-1 framework; not to mention, second order preserving time accurate feature, zero-order overshoot behaviour, unconditional stability, and with the least computational expense due to solving the system of equations only once and requiring the solutions of only one previous time step. Another significant contribution presented in this thesis was the development of an isochronous integration framework, which stemmed from the novel relations between the newly developed GS4-1 framework and the existing GS4-2 framework (for second order dynamic systems), to enable the use of the same computational framework for solving both the first and second order dynamic systems without having to re-

sort to the individual framework, hence the practicality in the computational aspects.

Additionally, this thesis also presented the numerical studies of silica particle deposition, where a numerical model of particle deposition was described, based on the well-founded theoretical framework, which was then used to gain more insights into the problem, and in particular into the effects of the flow and physicochemical parameters on the rate of the particle deposition, so that more effective measure for the mitigation of the problem can be further developed from such a study. This particle deposition problem is governed by partial differential equation that is first order in time and therefore suits the purpose of the development of the new time integration framework, which was the main focus in the present research. The results of the parametric studies undertaken based on the employed numerical model enable a few recommendations for the mitigation of the particle deposition problem, and therefore served as another valuable contribution of the thesis.

9.2 Future Directions

In this section, we briefly highlight the idea for future research involving the newly developed GS4-1 framework. Additionally, the idea of how the silica particle deposition model can be developed further in the future is also presented.

It is of our interest to investigate in the near future how the GS4-1 framework can be adapted to Differential-Algebraic Equations (DAE) systems that is first order in time such as those encountered in electrical engineering. A DAE is a generalization of an ordinary differential equation (ODE) in which the ODE is complemented by an algebraic constraint. Due to the presence of such a constraint, a system governed by a DAE exhibits fundamental mathematical properties that are different from those of ODE, and consequently poses additional challenges for their numerical solutions [130], which requires a thorough investigation and additional attention. Due to this reason, this type of systems is not considered in this thesis and will be addressed in future research. Such a research will provide enrichment to the applicability of the newly developed GS4-1 framework. It is also worth mentioning that, recently, the GS4-2 framework has been successfully applied to DAE systems,

particularly in application to the natural index 3 DAEs of multibody dynamics (second order systems) [131]. A precise understanding of the equation of motion time level concept as well as novel approaches for extending the basic primitive algorithm to nonlinear dynamic applications enabled a depth of search unique to the area. In the end, an algorithmic framework is identified which overcomes previous limitations and is capable of providing stable, robust, and accurate integration of index 3 DAEs for both rigid and rigid/flexible multibody dynamics applications [131]. Due to this breakthrough achievement, it is of future interest that an investigation is undertaken to adapt the GS4-1 framework to application in this type of systems (i.e., first order DAE). This will then provide a complete framework of isochronous integration method that is capable of providing stable, robust, and accurate integrations of both first and second order systems, that are governed by both partial differential equations and/or differential algebraic equations.

The particle deposition model presented in Chapter 8, considers, as a preliminary study, the deposition of colloidal particles from laminar, isothermal flow, onto walls of parallel plate channel. The model can be developed further in the future to include other processes that may appear in the deposition problem, such as

- turbophoresis: the transport of particles entrained in high velocity turbulent flow axially in the direction of decreasing turbulence level toward the wall region due to the large vertical gradient of turbulence near the depositing surface,
- thermophoresis: the transport of particles suspended in high temperature flow towards wall surface due to the presence of temperature gradient

Considering these two additional transport processes, which may greatly increase the rate of deposition of the particles, the problem to be solved is then governed by additional equations. In a nutshell, the governing equations describing the problem are

1. continuity equation
2. momentum equation

3. equation of turbulent energy
4. equation of turbulent dissipation
5. equation of energy
6. equation of particle transport and deposition

Equations (1) to (5) above are to be solved simultaneously to give the velocity field, pressure, enthalphy, turbulent energy, and turbulent dissipation of the flow. Using the information on the pressure and enthalpy, thermodynamic database can be called upon to give information on the temperature of the suspension. Assuming dilute particle suspension, in which the fluid flow field is not affected by the presence of the particles (one-way coupling), then the particle transport equation is then solved separately using the obtained information on the velocity field, temperature and turbulent characteristics. Additionally, the dynamics of the deposition process can be captured by looking at the evolution of the deposition interface, which can be tracked using a level set method. Such a complex computational procedure can be illustrated as a big picture in Figure 9.1.

The model can futher be developed to consider two-way coupling in which case not only the particles suspension is affected by the flow velocity field, but also vice versa. In this case, the particle transport equation cannot be solved separately, but instead it is solved simulataneously with all the other governing equations, which is a more complicated situation than that described above. In this case, it is recommended that the governing equations involved be implemented and solved using a Computational Fluid Dynamics package (CFD) due to the advantage of user-defined functions that can be programmed and implemented in most CFD softwares. In a nutshell, the CFD model will include a hydrodynamic module including turbulence, a heat transfer module, a particle transport module, and a fluid-structure interaction module enabling the investigation of interaction between the growth of the silica scale layer and the flow. Additionally, the particle transport module may also include monomeric type of deposition in addition to the colloidal deposition investigated in the present study. This opens up the prospects of a holistic model of silica scale deposition, which will

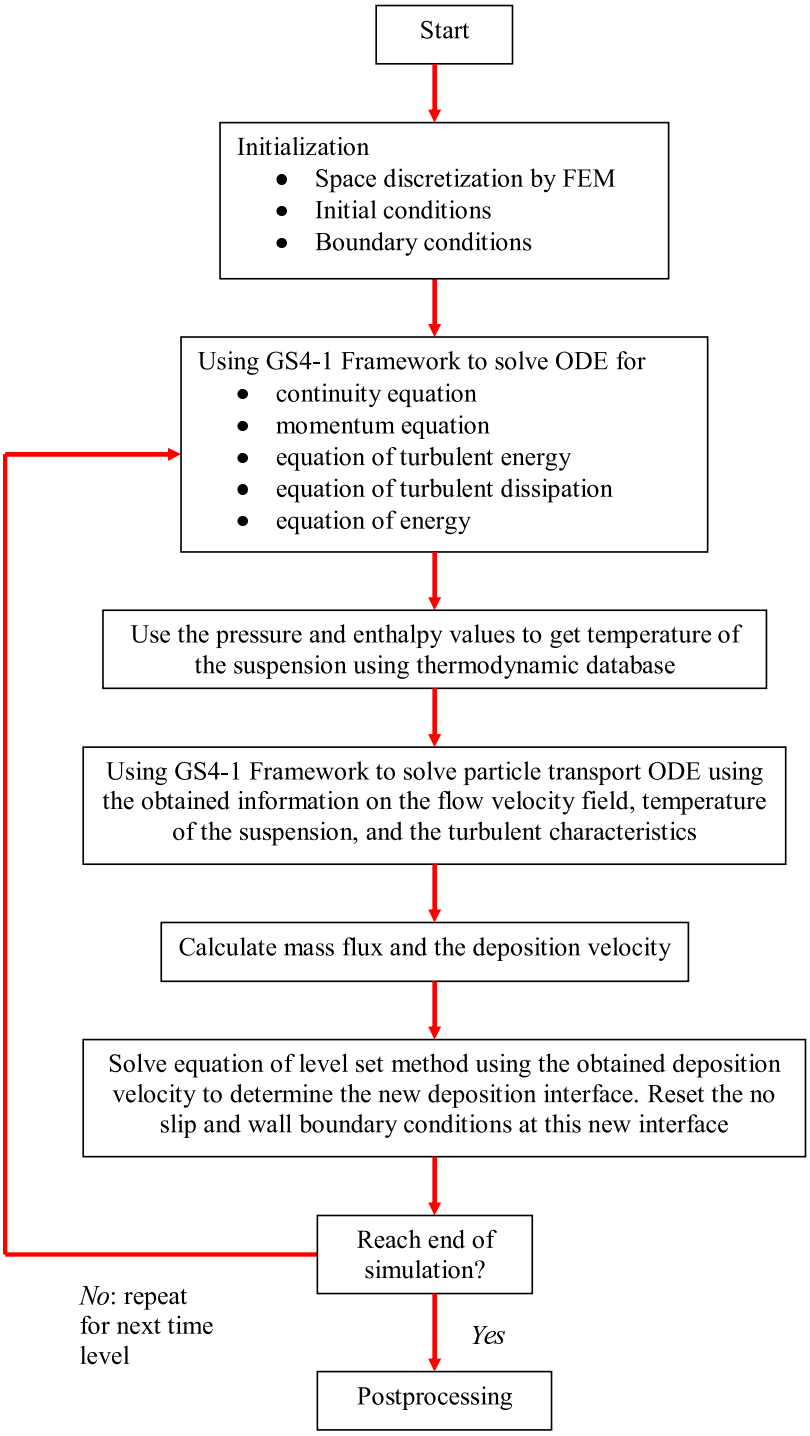


Figure 9.1: Computational procedure for solving particle deposition from turbulent flow subject to thermophoresis

ultimately guide geothermal engineers and geochemists in their endeavour to mitigate silica scale deposition more efficiently.

Bibliography

[1] A. Hoitink, S. Masuri, X. Zhou, and K.K. Tamma. On the Precise Evaluation of Acceleration Computations for General Structural Dynamic Applications: Issues and Noteworthy Perspectives. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 16th AIAA/ASME/AHS Adaptive Structures Conference; 10th AIAA Non-Deterministic Approaches Conference; 9th AIAA Gossamer Spacecraft Forum and the 4th AIAA Multidisciplinary Design Optimization Specialist Conference*, Schaumburg, IL, 7-10 April 2008.

[2] F.S. Loureiro and Webe Joao Mansur. A new family of time integration methods for heat conduction problems using numerical greens functions. *Computational Mechanics*, 44(4):519531, 2009.

[3] H. Brenner. The slow motion of a sphere through a viscous fluid towards a plane surface. *Chemical Engineering Science*, 16:242–251, 1961.

[4] A.J. Goldman, R.G. Cox, and H. Brenner. Slow viscous motion of a sphere parallel to a plane wall I. Motion through a quiescent fluid. *Chemical Engineering Science*, 22:637–651, 1967.

[5] A.J. Goldman, R.G. Cox, and H. Brenner. Slow viscous motion of a sphere parallel to a plane wall II. Couette flow. *Chemical Engineering Science*, 22:653–660, 1967.

[6] E. Bart. The slow unsteady settling of a fluid sphere towards a flat fluid interface. *Chemical Engineering Science*, 23:193–210, 1968.

[7] S.L. Goren and M.E. O'Neil. On the hydrodynamic resistance to a particle of

a dilute suspension when in the neighbourhood of a large obstacle. *Chemical Engineering Science*, 26:325–338, 1971.

[8] P. Warszynski. Coupling of hydrodynamic and electric interactions in adsorption of colloidal particles. *Advances in Colloid and Interface Science*, 84:47–142, 2000.

[9] J. Sjollema and H.J. Busscher. Deposition of Polystyrene Latex Particles toward Polymethylmethacrylate in a Parallel Plate Flow Cell. *Journal of Colloid and Interface Science*, 132(2):382–394, 1989.

[10] N.M. Newmark. A method of computation for structural dynamics. *Journal American Society of Civil Engineers*, 1:67–94, 1959.

[11] O.C. Zienkiewicz. A new look at Newmark, Houbolt and other time stepping formulas: a weighted residual approach. *Earthquake Engineering and Structural Dynamics*, 5:283, 1977.

[12] O.C. Zienkiewicz, W.L. Wood, N.W. Hine, and R.L. Taylor. A unified set of single-step algorithms, Part 1: general formulations and applications. *International Journal for Numerical Methods in Engineering*, 20:1529, 1984.

[13] W.L. Wood. A unified set of single-step algorithms, Part 2. *International Journal for Numerical Methods in Engineering*, 20:2303, 1984.

[14] K.K. Tamma and R.R. Namburu. Applicability and evaluation of an implicit self-starting unconditionally stable methodology for the dynamics of structures. *Computers and Structures*, 34.

[15] K.K. Tamma D. Sha, X. Zhou. Time discretized operators: towards the theoretical design of a new generation of implicit and explicit representations of time operators for structural dynamics: Part 2 . *Computer Methods in Applied Mechanics and Engineering*, 192:291–329, 2003.

[16] X. Zhou and K.K. Tamma. Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *International Journal for Numerical Methods in Engineering*, 59:597–668, 2004.

[17] E.L. Wilson. A computer program for dynamic stress analysis of underground structures. Technical report, SESM: University California, Berkeley, 1968.

[18] G.L. Goudreau and R.L. Taylor. Evaluation of numerical methods in elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 2:69, 1973.

[19] J.H. Argyris, P.C. Dunne and T. Angelopoulos. Nonlinear oscillations using the finite element technique. *Computer Methods in Applied Mechanics and Engineering*, 2:203, 1973.

[20] C. Hoff and P.J. Pahl. Development of an implicit method with numerical dissipation from a generalized single step algorithm for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 67:367, 1988.

[21] R.L. Taylor H.M. Hilber, TJR. Hughes. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5:283, 1977.

[22] H.M. Hilber. *Analysis and Design of numerical integration methods in structural dynamics*. PhD thesis, University of California, Berkeley, California, 1977.

[23] O.C. Zienkiewicz W.L. Wood, M. Bossak. An alpha modification of Newmark's method. *International Journal for Numerical Methods in Engineering*, 15:1562, 1980.

[24] H. P. Shao. The Studying on the Direct Time Integration Algorithms for Structural Dynamics Response. Master's thesis, Zheng Jiang University, 1987.

[25] H. P. Shao and C. W. Cai. The Direct Integration Three-Parameters Optimal Schemes for Structural Dynamics. In *Proceeding of the International Conference: Machine Dynamics and Engineering Applications*, pages C16–C20. Xi'an Jiaotong University Press, 1988.

[26] J. Chung and G. Hulbert. A time integration method for structural dynamics with improved numerical dissipation: the generalized α-method. *Journal of Applied Mechanics*, 30:371, 1993.

[27] X. Zhou and K.K. Tamma. Algorithms by design with illustrations to solid and structural mechanics/dynamics. *International Journal for Numerical Methods in Engineering*, 66:1738–1790, 2006.

[28] K.K. Tamma, X. Zhou, and D. Sha. The time dimension: a theory of development/evolution, classification, characterization and design of computational algorithms for transient/dynamic applications. *Archives in Computational Mechanics*, 7:67–290, 2000.

[29] X. Zhou, K.K. Tamma, and D. Sha. A theory of development and design of generalized integration operators for computational structural dynamics. *International Journal for Numerical Methods in Engineering*, 50:1619–1664, 2000.

[30] X. Zhou and K.K. Tamma. A new unified theory underlying time dependent linear first-order systems: a prelude to algorithms by design. *International Journal for Numerical Methods in Engineering*, 60:1699–1740, 2004.

[31] TJR. Hughes. Stability, convergence and growth and decay of energy of the average acceleration method in nonlinear structual dynamics. *Computers and Structures*, 6:313–324, 1976.

[32] T. Belytschko and D.F. Schoeberle. On the unconditional stability of an implicit algorithm for nonlinear structural dynamics. *Journal of Applied Mechanics*, 42:865–869, 1975.

[33] TJR. Hughes. A note on the stability of Newmark's algorithm in nonlinear structural dynamics. *International Journals for Numerical Methods in Engineering*, 11(2):383–386, 1977.

[34] J.C. Simo and N. Tarnow. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *Journal of Applied Mathematics and Physics*, 43:757, 1992.

[35] J.C. Simo and N. Tarnow. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics and Engineering*, 100:63, 1992.

[36] TJR. Hughes, T.K. Caughy, and W.K. Liu. Finite-element methods for nonlinear elastodynamics which conserve energy. *Journal of Applied Mechanics*, 45:366–370, 1978.

[37] D. Kuhl and E. Ramm. Constraint energy momentum algorithm and its application to nonlinear dynamics of shell. *Computer Methods in Applied Mechanics and Engineering*, 136:293–315, 1996.

[38] R.A. LaBudde and D. Greenspan. Energy and Momentum Conserving Methods of Arbitrary Order for the Numerical Integration of Equations of Motion. *Numer. Math.*, 25:323, 1976.

[39] W.L. Wood and M.E. Odour. Stability properties of some algorithms for the solution of nonlinear dynamic vibration equations. *Communications in Applied Numerical Methods*, 4:205–212, 1988.

[40] M.A. Crisfield and U. Galvanetto. Some unreliable finite element solutions for nonlinear dynamics. *Proc. 5th International Conference on Reliability of Finite Element Methods for Engineering Applications, Amsterdam, The Netherlands*, pages 346–355, 10-12 May, 1995.

[41] S.U. Masuri, A. Hoitink, X. Zhou, and K.K. Tamma. Algorithms by design: A new normalized time-weighted residual methodology and design of a family of energy-momentum conserving algorithms for non-linear structural dynamics. *International Journal of Numerical Methods in Engineering*, 79:1094–1146, 2009.

[42] S.U. Masuri. A Normalized Time Weighted Residual Approach for Nonlinear Structural Dynamics. Master's thesis, University of Minnesota, Minneapolis, Minnesota, 2007.

[43] S. Masuri, A. Hoitink, X. Zhou, and K.K. Tamma. Algorithms by design: Part II - A novel normalized time-weighted residual methodology and design leading to a family of symplectic-momentum conserving algorithms for nonlinear structural dynamics. *International Journal for Computational Methods in Engineering Science and Mechanics*, 10:27–56, 2009.

[44] S. Masuri, A. Hoitink, X. Zhou, and K.K. Tamma. Algorithms by Design: Part III A novel normalized time weighted residual methodology and design of optimal symplectic-momentum based controllable numerical dissipative algorithms for nonlinear structural dynamics. *International Journal for Computational Methods in Engineering Science and Mechanics*, 10:57–90, 2009.

[45] Z. Ge and J.E. Marsden. Lie-Poisson integrators and Lie-Poisson Hamilton-Jacobi theory. *Phys. Lett.*, A 133:134, 1988.

[46] D. Kuhl and M.A. Crisfield. Energy-conserving and decaying algorithms in non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 45:569–599, 1999.

[47] K. Feng. Difference Schemes for Hamiltonian Formalism and Symplectic Geometry. *Journal of Computational Mathematics*, 4:279–289, 1986.

[48] A.P. Veselov. Integrable Discrete-Time Systems and Difference Operators. *Functional Analysis and Its Applications*, 22:83–93, 1988. Translated from Funktsional'nyi Analiz i Ego Prilozheniya, vol. 22, pp. 1-13, 1988.

[49] J.E. Marsden C. Kane and M. Ortiz. Symplectic-energy-momentum preserving variational integrators. *Journal of Mathematical Physics*, 40:3353–3371, 1999.

[50] M. Ortiz C. Kane, J.E. Marsden and M. West. Variational Integrators and the Newmark Algorithm for Conservative and Dissipative Mechanical Systems. *International Journal for Numerical Methods in Engineering*, 49:1295–1325, 2000.

[51] J.E. Marsden and M. West. Discrete Mechanics and Variational Integrators. *Acta Numerica*, 10:357–514, 2001.

[52] L. Stainier L. Noels and J.P. Ponthot. An Energy-Momentum Conserving Algorithm for Non-Linear Hypoelastic Constitutive Models. *International Journal of Numerical Methods in Engineering*, 59:83–114, 2004.

[53] K.K. Tamma and R.R. Namburu. A New Finite Element Based Lax-Wendroff/Taylor-Galerkin Methodology for Computational Dynamics. *Computer Methods in Applied Mechanics and Engineering*, 71:137, 1988.

[54] A.P. Veselov. Integrable Lagrangian Correspondences and Factorization of Matrix Polynomials . *Functional Analysis and Its Applications*, 25:112–122, 1991. Translated from Funktsional'nyi Analiz i Ego Prilozheniya, vol. 25, pp. 38-49, 1991.

[55] J. Moser and A.P. Veselov. Discrete Versions of Some Classical Integrable Systems and Factorization of Matrix Polynomials. *Communications in Mathematical Physics*, 139:217–243, 1991.

[56] R.S. MacKay. Some Aspects of the Dynamics and Numerics of Hamiltonian System. In D. S. Broomhead and A. Iserles, editors,. *The Dynamics of Numerics and the Numerics of Dynamics*. pages 137-193. Oxford University Press, Oxford, 1992.

[57] J.C. Simo. Nonlinear Stability of the Time-Discrete Variational Problem of Evolution in Nonlinear Heat Conduction, Plasticity and Viscoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 88:111–131, 1991.

[58] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Springer, New York, 1998.

[59] V. Wheeler, S. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. On the applicability of an isochronous integration framework for parabolic/hyperbolic heat conduction type problems. *Journal of Numerical Heat Transfer*.

[60] J. Crank and P. Nicolson. Practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43:50–67, 1947.

[61] E.A. Thornton K.H. Huebner and T.G. Byrom. *The Finite Element Method for Engineers*. John Wiley & Sons, Inc, New York, 3 edition, 1995.

[62] K.E. Jansen, C.H. Whiting, and G.M. Hulbert. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.

[63] C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[64] A. Belmonte and A. Libchaber. Thermal signature of plumes in turbulent convection: The skewness of the derivative. *Phys. Rev. E*, 53:4893–4898, May 1996.

[65] J. Schroers, A. Masuhr, W.L. Johnson, and R. Busch. Pronounced asymmetry in the crystallization behavior during constant heating and cooling of a bulk metallic glass-forming liquid. *Phys. Rev. B*, 60:11855–11858, Nov 1999.

[66] S. Masuri, M. Sellier, X. Zhou, and K.K. Tamma. Design of order-preserving algorithms for transient first-order systems with controllable numerical dissipation. *International Journal for Numerical Methods in Engineering*, 88:1411–1448, 2011.

[67] S. Masuri, K.K. Tamma, X. Zhou, and M. Sellier. GS4-1 computational framework for heat transfer problems: Part 1 - linear cases, with illustration to thermal shock problem. *Journal of Numerical Heat Transfer*, 2012. to appear.

[68] S. Masuri, K.K. Tamma, X. Zhou, and M. Sellier. GS4-1 computational framework for heat transfer problems: Part 2 - extension to nonlinear cases, with illustrations to radiation heat transfer problem. *Journal of Numerical Heat Transfer*, 2012. to appear.

[69] B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, New York, 1972.

[70] TJR. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[71] G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3:27, 1963.

[72] J.N. Franklin. *Matrix Theory*. Prentice-Hall, Englewood Cliffs, NJ, 1968.

[73] A. Hoitink, S. Masuri, X. Zhou, and K.K. Tamma. Algorithms by Design: Part I - On the Hidden Point Collocation Within LMS methods and Implications for Nonlinear Dynamics Applications. *International Journal for Computational Methods in Engineering Science and Mechanics*, 9:383–407, 2008.

[74] M.M. Grigoriev and G.F. Dargush. Boundary element methods for transient convective diffusion. Part III: Numerical examples. *Computer methods in applied mechanics and engineering*, 192:4313–4335, 2003.

[75] A. Durmus, I. Boztosun and F. Yasuk. Comparative study of the multiquadric and thin-plate spline radial basis functions for the transient-convective diffusion problems. *International Journal of Modern Physics C*, 17(8):1151–1169, 2006.

[76] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[77] H.S. Carslaw and J.C. Jaeger. *Conduction of heat in solids*. Oxford University Press, Oxford, 1959.

[78] E.R. Benton and G.W. Platzman. A table of solutions of the one-dimensional Burgers equation. *Quart. Appl. Math.*, 30:195–212, 1972.

[79] A.R. Bahadir. A fully implicit finite-difference scheme for two-dimensional Burgers' equations. *Applied Mathematics and Computation*, 137:131–137, 2003.

[80] R.D. Krieg. Unconditional stability of numerical time integration methods . *Journal of Applied Mechanics*, 40:417–421, 1973.

[81] K.K. Tamma and R.R. Namburu. Computational approaches with applications to non-classical and classical thermomechanical problems. *Applied Mechanics Reviews*, 50(9):514–551, 1997.

[82] J.L. Nowinski. *Theory of thermoelasticity with applications*. Sijthoff & Noordhoff International Publishers, The Netherlands, 1978.

[83] V.I. Danilovskaya. Thermal Stresses in an Elastic Half-Space due to a Sudden Heating of its Boundary. *(in Russian), Prikladnaya Matematika i Mekhanika*, 14:316–318, 1950.

[84] E. Sternberg and J.G. Chakravorty. On inertia effects in a transient thermoelastic problem. *Journal of Applied Mechanics*, 26:503–509, 1959.

[85] M. Balla. Analytical study of the thermal shock problem of a half-space with various thermoelastic models. *Acta Mechanica*, 89:73–92, 1991.

[86] Ministry of Economic Development. (2011, 04-01-2012). New Zealand Energy Data File. Available: http://www.med.govt.nz/sectors-industries/energy/energy-modelling/publications/energy-data-file/new-zealand-energy-data-file-2011.

[87] Mighty River Power. (2010, 02-12-2011). Geothermal Generation. Available: http://www.mightyriverpower.co.nz/Generation/PowerStations/Geothermal/.

[88] R.C. Schroeder. Modeling the temperature-dependent scale accumulation from geothermal brine. Technical report, Lawrence Livermore Laboratory, University of California, 1976.

[89] J.S. Gudmundsson and T.R. Bott. Deposition of Silica From Geothermal Waters on Heat-Transfer Surfaces. *Desalination*, 28(2):125–145, 1979.

[90] R.E. Jamieson. Geothermal Silica Scaling. Master's thesis, University of Auckland, 1984.

[91] S.K. Friedlander and H.F. Johnstone. Deposition of Suspended Particles from Turbulent Gas Streams. *Industrial and Engineering Chemistry*, 49(7):1151–1156, 1957.

[92] A. Guha. Transport and Deposition of Particles in Turbulent and Laminar Flow. *Annual Review of Fluid Mechanics*, 40(1):311–341, 2008.

[93] P.J. Fryer. Modelling the behaviour of heat exchangers undergoing scaling. *Geothermics*, 18(1/2):89–96, 1989.

[94] P.J. Fryer and N.K.H Slater. Reaction fouling from food fluids. In *Fouling of Heat Exchanger Plant*, pages 65–73. ASME HTD-35, 1984.

[95] D.Q. Kern and J.G. Seaton. A theoretical analysis of thermal surface fouling. *British Chemical Engineering*, 4(5):258–262, 1959.

[96] J. Pott, M.G. Dunstall, K.L. Brown. Numerical simulation of silica scaling. In *Proceedings 18th NZ Geothermal Workshop*, pages 41–46, 1996.

[97] T.G.M. van de Ven. *Colloidal Hydrodynamics*. Academic Press, London, 1989.

[98] S.H. Chan and B. Moussa. Trajectories and deposition of silica particles on cylinders in crossflow with and without a magnetic field. *Journal of Heat Transfer-Transactions of the Asme*, 118(4):903–910, 1996.

[99] V.G. Levich. *Physicochemical Hydrodynamics*. Prentice Hall, New Jersey, 1962.

[100] Z. Adamczyk and P. Weronski. Application of the DLVO theory for particle deposition problems. *Advances in Colloid and Interface Science*, 83:137–226, 1999.

[101] Z. Adamczyk and T.G.M van de Ven. Deposition of Brownian particles onto cylindrical collectors. *Journal Of Colloid And Interface Science*, 84:497–518, 1981.

[102] Z. Adamczyk and T.G.M van de Ven. Deposition of particles under external forces in laminar flow through parallel-plate and cylindrical channels. *Journal Of Colloid And Interface Science*, 80:340–356, 1981.

[103] Z. Adamczyk and T.G.M van de Ven. Particle transfer to a plate in uniform flow. *Chemical Engineering Science*, 37:869–880, 1982.

[104] S.L. Soo. Pipe Flow of Suspensions. *Applied Scientific Research*, 21(1):68, 1969.

[105] S.L. Soo and S.K. Tung. Pipe Flow of Suspensions in Turbulent Fluid - Electrostatic and Gravity Effects. *Applied Scientific Research*, 24(2-3):83, 1971.

[106] M. Elimelech, J. Gregory, X. Jia, and R.A. Williams. *Particle Deposition and Aggregation: Measurement, Modelling and Simulation.* Butterworth-Heinemann, Woburn, Massachusetts, 1995.

[107] J. Mahanty and B.V. Ninham. *Dispersion Forces.* Academic Press, New York, 1976.

[108] J.N. Israelachvili. *Intermolecular and Surface Forces.* Academic Press, London, 1985.

[109] R.J. Hunter. *Foundations of Colloid Science.* V.I. Clarendon Press, Oxford, 1986.

[110] Z. Adamczyk, J. Czarnecki, T. Dabros, and T.G.M van de Ven. Particle transfer to solid surfaces. *Advances in Colloid and Interface Science*, 19:183, 1983.

[111] L. Song and M. Elimelech. Calculation Of Particle Deposition Rate Under Unfavorable Particle Surface Interactions. *Journal Of The Chemical Society-Faraday Transactions*, 89(18):3443–3452, 1993.

[112] Z. Adamczyk and T.G.M van de Ven. Kinetics of particle accumulation at collector surfaces I. Approximate analytical solutions. *Journal Of Colloid and Interface Science*, 97:68–90, 1984.

[113] M. Elimelech. Particle Deposition On Ideal Collectors From Dilute Flowing Suspensions - Mathematical Formulation, Numerical-Solution, And Simulations. *Separations Technology*, 4(4):186–212, 1994.

[114] M. Elimelech and C.R. Omelia. Effect Of Particle-Size On Collision Efficiency In The Deposition Of Brownian Particles With Electrostatic Energy Barriers. *Langmuir*, 6(6):1153–1163, 1990.

[115] M. Elimelech. Effect Of Particle-Size On The Kinetics Of Particle Deposition Under Attractive Double-Layer Interactions. *Journal Of Colloid And Interface Science*, 164(1):190–199, 1994.

[116] E. Ruckenstein and D.C. Prieve. Rate of Deposition of Brownian Particles under Action of London and Double-Layer Forces. *Journal of the Chemical Society-Faraday Transactions II*, 69(10):1522–1536, 1973.

[117] L.A. Spielman and S.K. Friedlander. Role of Electrical Double-Layer in Particle Deposition by Convective Diffusion. *Journal of Colloid and Interface Science*, 46(1):22–31, 1974.

[118] B.D. Bowen, S. Levine and N. Epstein . Fine Particle Deposition in Laminar Flow Through Parallel-Plate and Cylindrical Channels. *Journal of Colloid and Interface Science*, 54(3):375–390, 1976.

[119] Z. Adamczyk and T. Dabros. Convective Diffusion of Particles under Electrical Double-Layer Forces. *Journal Of Colloid And Interface Science*, 64(3):580–583, 1978.

[120] T. Dabros and Z. Adamczyk. Noninertial Particle Transfer to the Rotating-Disk under an External Force-Field (Laminar Flow). *Chemical Engineering Science*, 34(8):1041–1049, 1979.

[121] M. Elimelech. Predicting Collision Efficiencies Of Colloidal Particles In Porous-Media. *Water Research*, 26(1):1–8, 1992.

[122] B.D. Bowen and N. Epstein . Fine Particle Deposition in Smooth Parallel-Plate Channel. *Journal of Colloid and Interface Science*, 72(1):81–97, 1979.

[123] P.R. Johnson and M. Elimelech. Dynamics of Colloid Deposition in Porous Media: Blocking Based on Random Sequential Adsorption. *Langmuir*, 11:801–812, 1996.

[124] C-H. Ko, S. Bhattacharjee, and M. Elimelech. Coupled Influence of Colloidal and Hydrodynamic Interactions on the RSA Dynamic Blocking Function for Particle Deposition onto Packed Spherical Collectors. *Journal of Colloid and Interface Science*, 229:554–567, 2000.

[125] Z. Adamczyk and P. Weronski. Kinetics of Irreversible Adsorption of Interacting Spheroidal Particles. *Langmuir*, 11(11):4400–4410, 1995.

[126] Z. Adamczyk, M. Zembala, B. Siwek, and P. Warszynski. Structure and Ordering in Localized Adsorption of Particles. *Journal Of Colloid and Interface Science*, 140(1):123–137, 1990.

[127] E.J.W. Verwey. Theory of the Stability of Lyophobic Colloids. *J. Phys. Chem.*, 51(3):631636, 1947.

[128] E.J.W. Verwey and J.Th.G. Overbeek. *Theory of the Stability of Lyophobic Colloids*. Elsevier, Amsterdam-New York, 1948.

[129] P. Adomeit and U. Renz. Deposition of Fine Particles from a Turbulent Liquid Flow: Experiments and Numerical Predictions. *Chemical Engineering Science*, 51(13):3491–3503, 1996.

[130] S.L. Campbell, Vu Hoang Linh, and L.R. Petzold. Differential-algebraic equations. 3(8):2849, 2008.

[131] A.J. Hoitink. *Application of the GSSSS Family of Algorithms to the Natural Index 3 Differential-Algebraic Equations of Multibody Dynamics*. PhD thesis, University of Minnesota, Minneapolis, Minnesota, 2011.